# NAVAL POSTGRADUATE SCHOOL

**MONTEREY, CALIFORNIA**

# THESIS

**ACHIEVING SINK NODE ANONYMITY UNDER ENERGY CONSTRAINTS IN WIRELESS SENSOR NETWORKS**

by

Audrey F. Callanan

June 2014

| | |
|---|---|
| Thesis Advisor: | Preetha Thulasiraman |
| Second Reader: | Mark Gondree |

**Approved for public release; distribution is unlimited**

THIS PAGE INTENTIONALLY LEFT BLANK

## REPORT DOCUMENTATION PAGE

*Form Approved OMB No. 0704-0188*

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington DC 20503.

|  |  |  |
|---|---|---|
| **1. AGENCY USE ONLY** ***(Leave blank)*** | **2. REPORT DATE**<br>June 2014 | **3. REPORT TYPE AND DATES COVERED**<br>Master's Thesis |

|  |  |
|---|---|
| **4. TITLE AND SUBTITLE**<br>ACHIEVING SINK NODE ANONYMITY UNDER ENERGY CONSTRAINTS IN WIRELESS SENSOR NETWORKS | **5. FUNDING NUMBERS** |
| **6. AUTHOR(S)** Audrey F. Callanan |  |
| **7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)**<br>Naval Postgraduate School<br>Monterey, CA 93943-5000 | **8. PERFORMING ORGANIZATION REPORT NUMBER** |
| **9. SPONSORING /MONITORING AGENCY NAME(S) AND ADDRESS(ES)**<br>N/A | **10. SPONSORING/MONITORING AGENCY REPORT NUMBER** |

**11. SUPPLEMENTARY NOTES** The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government. IRB Protocol number ____N/A____.

|  |  |
|---|---|
| **12a. DISTRIBUTION / AVAILABILITY STATEMENT**<br>Approved for public release; distribution is unlimited | **12b. DISTRIBUTION CODE**<br>A |

**13. ABSTRACT (maximum 200 words)**

A wireless sensor network (WSN) is a distributed network that facilitates wireless information gathering within a region of interest. For this reason, WSNs are relied upon by the Department of Defense for deployment in remote and hostile areas. The information collected by sensors is aggregated at a central point known as a sink node. Two challenges in the deployment of WSNs are limited battery power of each sensor node and sink node privacy/anonymity. The role played by the sink node raises its profile as a high value target for attack, thus its anonymity is crucial to the security of a WSN. In order to improve network security, we must implement a protocol that conceals the sink node's location while being cognizant of energy resource constraints. In this thesis, we develop a routing algorithm based on node clustering to improve sink node anonymity while simultaneously limiting node energy depletion. Via MATLAB simulations, we analyze the effectiveness of this algorithm in obfuscating the sink node's location in the WSN while preserving node energy. We show that the anonymity of the sink node is independent of traffic volume and that the average energy consumed by a node remains consistent across topological variations.

|  |  |
|---|---|
| **14. SUBJECT TERMS** Wireless Sensor Networks, WSN, Ad Hoc Network, ground sensor network, Cluster routing, Sink Node Anonymity, Base Station Anonymity, Location Privacy | **15. NUMBER OF PAGES**<br>149 |
|  | **16. PRICE CODE** |

|  |  |  |  |
|---|---|---|---|
| **17. SECURITY CLASSIFICATION OF REPORT**<br>Unclassified | **18. SECURITY CLASSIFICATION OF THIS PAGE**<br>Unclassified | **19. SECURITY CLASSIFICATION OF ABSTRACT**<br>Unclassified | **20. LIMITATION OF ABSTRACT**<br>UU |

THIS PAGE INTENTIONALLY LEFT BLANK

# ACHIEVING SINK NODE ANONYMITY UNDER ENERGY CONSTRAINTS IN WIRELESS SENSOR NETWORKS

Audrey F. Callanan
Captain, United States Marine Corps
B.S., United States Naval Academy, 2008

Submitted in partial fulfillment of the
requirements for the degree of

**MASTER OF SCIENCE IN ELECTRICAL ENGINEERING**

from the

**NAVAL POSTGRADUATE SCHOOL**
**June 2014**

Author: Audrey F. Callanan

Approved by: Preetha Thulasiraman, Ph.D.
Thesis Advisor

Mark Gondree, Ph.D.
Second Reader

Clark Robertson, Ph.D.
Chair, Department of Electrical and Computer Engineering

THIS PAGE INTENTIONALLY LEFT BLANK

# ABSTRACT

A wireless sensor network (WSN) is a distributed network that facilitates wireless information gathering within a region of interest. For this reason, WSNs are relied upon by the Department of Defense for deployment in remote and hostile areas. The information collected by sensors is aggregated at a central point known as a sink node. Two challenges in the deployment of WSNs are limited battery power of each sensor node and sink node privacy/anonymity. The role played by the sink node raises its profile as a high value target for attack, thus its anonymity is crucial to the security of a WSN. In order to improve network security, we must implement a protocol that conceals the sink node's location while being cognizant of energy resource constraints. In this thesis, we develop a routing algorithm based on node clustering to improve sink node anonymity while simultaneously limiting node energy depletion. Via MATLAB simulations, we analyze the effectiveness of this algorithm in obfuscating the sink node's location in the WSN while preserving node energy. We show that the anonymity of the sink node is independent of traffic volume and that the average energy consumed by a node remains consistent across topological variations.

THIS PAGE INTENTIONALLY LEFT BLANK

# TABLE OF CONTENTS

THIS PAGE INTENTIONALLY LEFT BLANK

# LIST OF FIGURES

# LIST OF TABLES

THIS PAGE INTENTIONALLY LEFT BLANK

# LIST OF ACRONYMS AND ABBREVIATIONS

| | |
|---|---|
| AvgEC | average energy consumed |
| CH | cluster head |
| DOD | Department of Defense |
| IEEE | Institute of Electrical and Electronics Engineers (IEEE) |
| GPS | Global Positioning System |
| LEACH | Low Energy Adaptive Cluster Hierarchy |
| LPR | Location Privacy Routing |
| MAC | Medium Access Control |
| MaxEC | maximum energy consumed |
| MaxECN | maximum energy consumed node |
| MinEC | minimum energy consumed |
| OSI | Open Systems Interconnect |
| PDF | Probabilistic Distribution Function |
| RRHA | Randomize Routing with Hidden Address |
| SPIN | Adaptive Protocols for Information Dissemination in WSN |
| USMC | United States Marine Corps |
| WSN | Wireless Sensor Network |

THIS PAGE INTENTIONALLY LEFT BLANK

# EXECUTIVE SUMMARY

Wireless sensor networks (WSNs) are ad-hoc networks in which sensor nodes are widely distributed in a region of interest for data extraction in real time. A sensor observes an event or gathers some physical data from its area of interest. It then processes the observed or gathered data using a tiny embedded processor. The sensor sends the processed data to a central data collector. The sensor nodes act as both sensing and routing devices. Multiple sensor nodes may be used to transmit data from the initial source node to the destination (i.e., multi-hop communication). The destination node in a WSN is characterized as the sink node.

When a WSN is deployed, each sensor has a finite amount of energy. Each action (i.e., sensing, transmitting etc.) that is taken by a sensor has an energy cost that slowly depletes the sensor's power. The death of a single node does not have a major impact on the WSN, but as additional nodes die out, the performance of the WSN is degraded.

WSNs greatly extend our ability to monitor and control the physical environment from remote locations and improve the accuracy of information obtained via collaboration among sensor nodes and online information processing at these nodes [1]. For this reason, WSNs are currently used for a broad range of military, civilian, and commercial applications.

WSN security is especially important from the DOD perspective; failure to protect the network can completely subvert the intended purpose of the sensor network [2]. These networks are remotely deployed and are vulnerable to malicious infiltration. It can no longer be assumed that an adversary has to be technologically advanced to observe or interfere with a deployed WSN. Due to the shared nature of wireless communication media, an attacker can easily eavesdrop on the radio communications either by purchasing their own sensor devices or by leveraging other radio devices capable of monitoring message transmission. The information that is revealed is meaningful-where the communication occurred and who participated in the communication.

The sink node in a WSN is crucial for gathering, aggregating, and transferring sensor information. Specific to DOD applications, the sink node is relied upon to provide critical information to personnel on the ground about an area of interest. The role played by the sink node in the WSN raises its profile as a high value target for attack. Since the sink node is a central point of failure, an adversary can destroy the sink and render ineffective the data gathering duties of the entire sensor network.

The privacy of the sink's location is a unique problem in WSNs. The protection of the sink's location cannot be achieved using existing security mechanisms, such as packet encryption, key management, etc. Therefore, it is important to develop and implement specific protocols that conceal the sink node's location. At the same time, a scheme for sink protection should not affect normal sensing and communication tasks that require knowledge of the sink's location. In most cases, sensed data is transmitted along paths from source nodes to a sink node. This produces pronounced traffic patterns that reveal the direction and, thus, the location of the sink node. An adversary can analyze the traffic patterns to deduce the location of the sink.

Another important parameter in achieving sink node anonymity/privacy is the issue of node energy maintenance. Any anonymity scheme that is implemented in a WSN must ensure that the energy of the nodes in not significantly depleted. Thus, a balance must be achieved in which sink node anonymity is attained while keeping node energy levels sufficient enough to continue network operations.

To address the issue of sink node privacy/anonymity, we develop a strategy to obfuscate the sink node's location using a hierarchical routing mechanism, known as clustering, while simultaneously limiting node energy depletion. To the best of our knowledge, this is the first work that develops a novel sink node anonymity algorithm in a resource efficient manner. The contributions of this thesis can be summarized as follows:

- Development and implementation of a network topology and clustering algorithm in a resource-efficient manner.
- Development of a routing algorithm for sink node anonymity.

- Simulation and evaluation of the routing algorithm for security robustness and energy preservation.

When the WSN is deployed the nodes are randomly distributed throughout the entire area of interest. The network model which is used in this thesis is a square 100 meter by 100 meter area. There are 100 nodes in the model. From a global view of the sensor area, the placements do not follow any pattern and, thus, can be modeled by a random distribution. The sink node is deliberately placed at the location ($x$,$y$)=(25 meters, 75 meters). The location of the sink node is deliberate because the personnel responsible for deploying the WSN deliberately place the sink node, likely co-locating it with their observation post.

Once the WSN deployed, the first step in our proposed algorithm is the initialization and formation of clusters. All of the nodes in the WSN either elect to become a Cluster Head (CH) or join a cluster as a cluster member, with the exception of the sink node. The sink node is always a cluster member in the WSN; it is never elected to be a CH. Each sensor in the WSN may elect to become a CH with a fixed probability $p$ when the network is deployed. An iterative approach is utilized to balance the competing demands of preventing isolation and achieving energy efficiency. In this thesis the probability of a sensor node electing to become a CH $p$ is fixed at 0.20. We choose three iterations to elect the CHs. At the end of the final iteration of CH election, all nodes in the WSN are either CHs or cluster members.

Let $N$ be the set of all nodes in the WSN and let $i$ denote the total number of nodes. In this thesis $i$=100 nodes. Now

$$N = \{n_1, n_2, .....n_i\}\,. \tag{1}$$

$CH$ is the set of nodes which serve as CHs. The total number of CHs is denoted as $j$:

$$CH = \{\mathrm{ch}_1, \mathrm{ch}_2, .....\mathrm{ch}_j\}\,. \tag{2}$$

$CM$ is the set of nodes which serve as cluster members. The total number of cluster members is denoted as $k$:

$$CM = \{\mathrm{cm}_1, \mathrm{cm}_2, .....\mathrm{cm}_k\}\,. \tag{3}$$

Each $n_i$ in $N$ becomes an element of $CH$ or $CM$:

$$n_i = ch_b \in CH \text{ or } n_i = cm_b \in CM \quad . \tag{4}$$

The goal of developing this algorithm is to ensure that at least *n* other nodes in the WSN have similar traffic statistics as the sink node.

Clustering also imposes a substantial energy burden on the nodes that act as CHs, therefore, we rotate the CHs. The CHs are rotated when one of two conditions are met. Either one of the CHs has expended a certain amount of energy or a specific number of messages have been transmitted through the WSN. Implementing CH rotation allows us to distribute the burden of being the CH across the WSN while increasing the overall lifetime of the WSN.

The CHs in the WSN are responsible for routing data from the source node's CH to the sink node's CH. When forwarding data to the next node, each CH has two options. The message can be directly forwarded to the next node or widely broadcast to all sensors within range. In this algorithm we choose a subset of CHs to broadcast. The sink node's CH always broadcasts the messages it receives so that the sink node can receive the information. The total number of broadcast CHs (denoted as BCCH in Eq. 5) is denoted as *m*:

$$BCCH = \{\mathrm{bc}_1, \mathrm{bc}_2, .....\mathrm{bc}_m\} \text{ and } BBCH \subseteq CH \, . \tag{5}$$

By broadcasting traffic to nodes other than the sink, we are essentially creating a situation where additional nodes resemble the sink in terms of traffic volume. In this thesis we determine that we would like a lower threshold of at least 20 nodes to have similar traffic statistics. In other words, from the adversary's perspective, there are multiple nodes acting like sink nodes. The number of nodes broadcast to directly correlates to the anonymity of the sink node. Once the threshold of 20 nodes is exceeded, no additional broadcast CHs are selected.

To choose the broadcast CH, the CHs are ordered by their residual energy levels:

$$CH_{energy} = \{\mathrm{ch}_5, \mathrm{ch}_8, .....\mathrm{ch}_j\} \, . \tag{6}$$

Each broadcast CH is selected in order of maximum energy remaining: $bc_1 = ch_5$, $bc_2 = ch_8$ and so on. A broadcast CH broadcasts any data it receives to all of its cluster members in addition to the next hop CH. The total number of nodes broadcast to is denoted as β:

$$\beta = \sum_{i=1}^{m} \text{members}(bc_i)\,. \tag{7}$$

The anonymity factor of the sink node is denoted as *AF* and is defined to be:

$$AF_{topo\text{logy}} = 1/\,average(\beta)\,. \tag{8}$$

Once broadcast CHs are determined, we determine the paths that traffic takes to reach the sink node’s CH. To establish routing paths, each CH uses Dijkstra’s routing algorithm to determine the path to the sink node’s CH. Dijkstra’s algorithm is a well-known, simple, least-cost algorithm that finds the lowest cost path from a source to a destination [3].  We used Euclidian distance as the cost between two CHs in Dijkstra’s algorithm. The resulting path is the most energy efficient route through the WSN without factoring in the additional cost of the broadcast CHs.

The algorithmic process discussed above is shown in Figures 1, 2, 3 and 4.

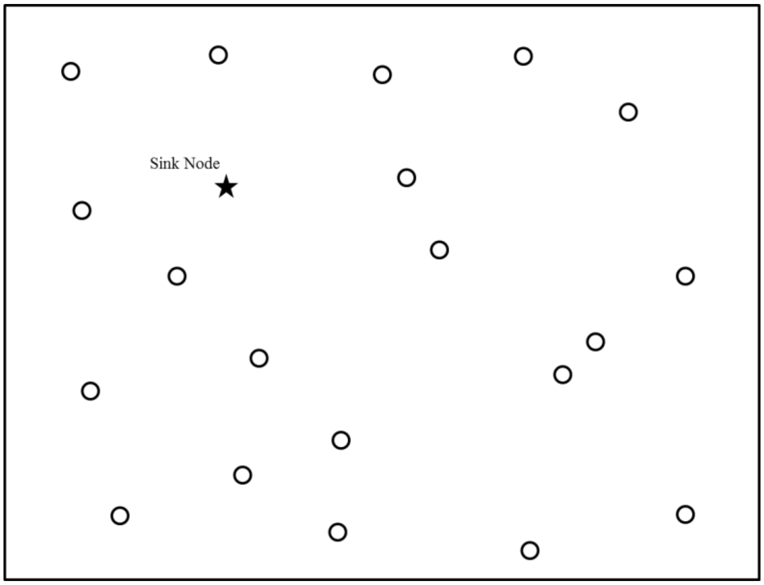


Figure 1. The WSN is deployed. All of the sensor nodes are placed randomly except the sink node, which is placed at (25 m, 75 m).

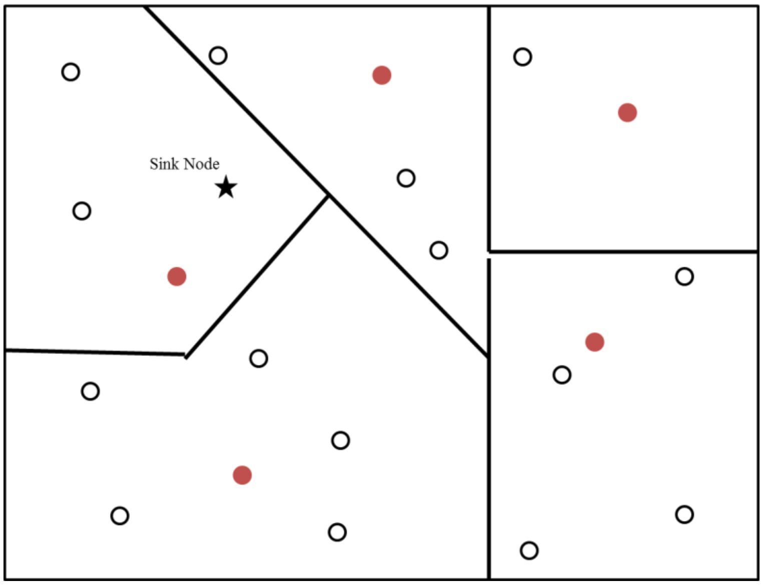


Figure 2. The WSN forms clusters with the election of CHs.

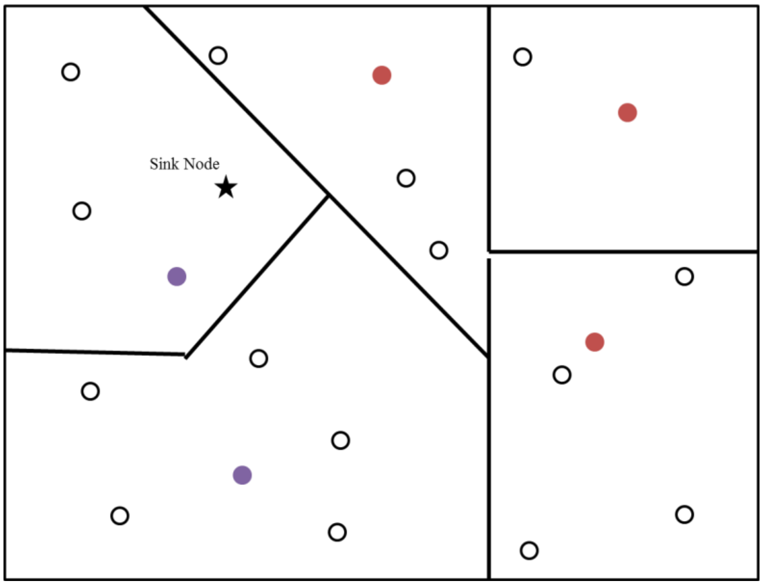


Figure 3. A subset of the CHs become broadcast CHs.

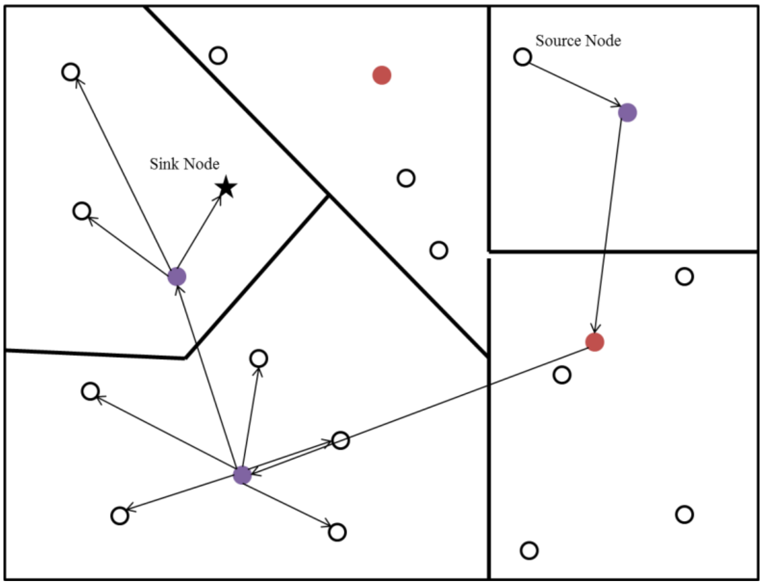


Figure 4. All cluster heads utilize Dijkstra's algorithm to determine the least cost route to the sink node's CH. Traffic is routing using the results of Dijkstra's algorithm. Broadcast CHs broadcast the data to all their cluster members.

In our simulations we generated four different topologies. Each topology represents a different physical location of the nodes in the WSN. We generated traffic to be routed across the WSN at four different traffic volumes: 5,000 messages, 10,000 messages, 15,000 messages and 20,000 messages. We conducted five trials at each traffic volume on each topology. For simplicity, we did not let any nodes die out in these simulations because when nodes die the WSN may become partitioned, making the problem more difficult. Our goal is to evaluate the performance of the algorithm over a network where all of the nodes are alive.

To evaluate the anonymity factor, we take the average value of the cluster members broadcast to across the simulation. From our simulations we were able to evaluate the resource efficiency and resulting sink node anonymity level of our proposed algorithm.

Considering all four topologies individually and averaged together, we find remarkably consistent results for the average amount of energy consumed by a node in the WSN, as shown in Figure 2. The average energy consumed by a node increases as the traffic volume increases for each topology. Comparing the results side by side on the same plot, we see that the average energy consumed by a node in Topology 1 is consistently less than the other topologies. We expect this variation among the topologies as the physical location of the nodes affects the energy consumption of each node in the WSN. These results are promising because the average energy use by each node is an effective parameter for planning overall network lifetime.

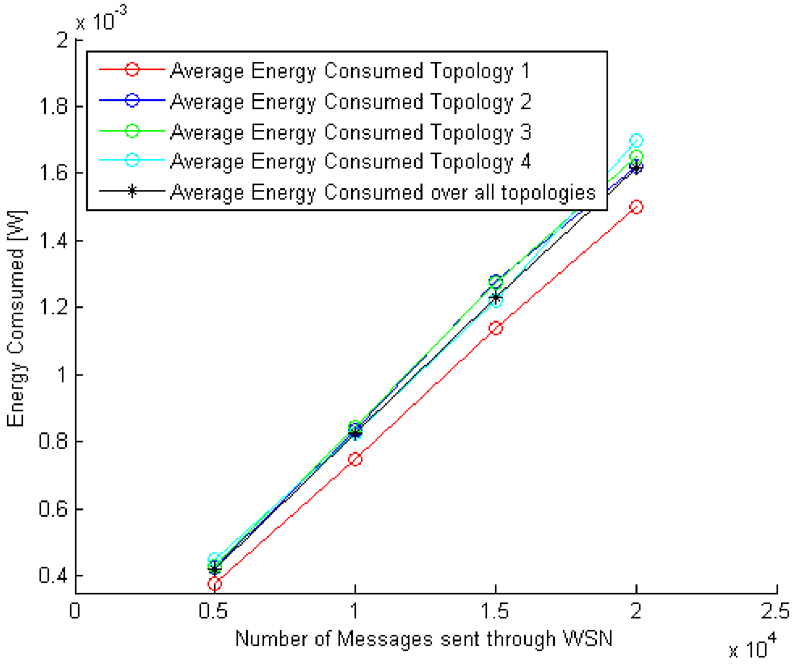


Figure 5. The average energy consumed by a node for all four topologies and the average of the four. The average energy consumed by a node in the WSN increases as the traffic volume through the WSN increases from 5,000 messages to 20,000 messages. The results are consistent across the four topologies simulated.

From Figure 3, it can be seen that the maximum energy consumed for all four topologies and their average. We can see that the maximum energy consumed by a node for each topology and traffic volume varies more than the average energy consumed. From Figure 3, we see that the maximum energy consumed by a node of the different topologies is not tightly grouped at any of the traffic volumes. We attribute the maximum energy consumed variations to many roles played by the node (i.e., cluster member, CH, and broadcast CH). Each of these roles contributes to the energy consumption of the node, and because each of these roles is randomized, the MaxEC is highly variable.

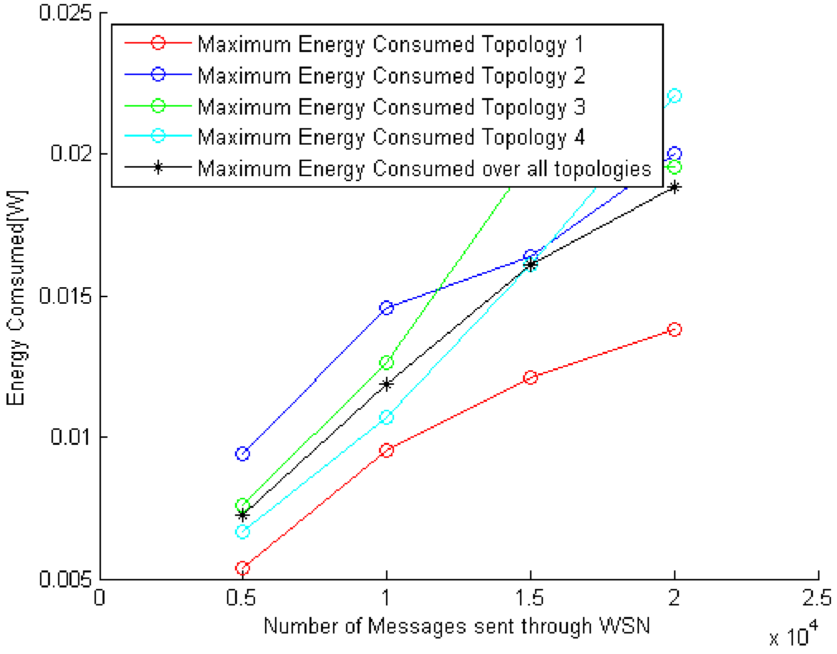
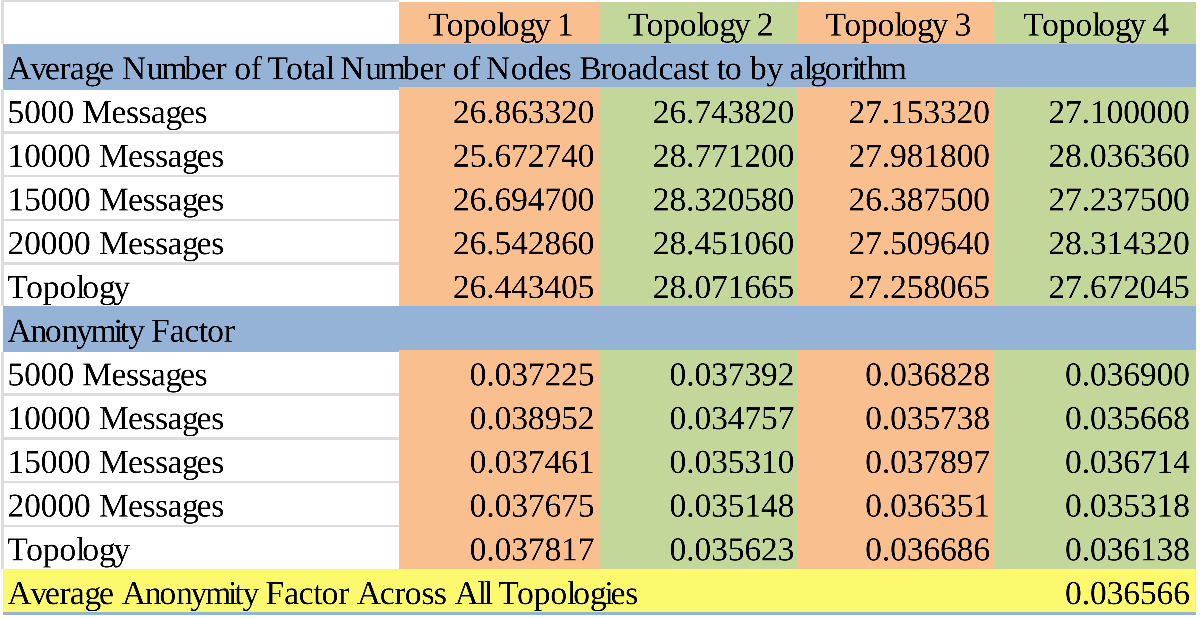


Figure 6. The maximum energy values for all four topologies and their average. The maximum energy consumed by a node in the WSN increases as the traffic volume through the WSN increase from 5,000 messages to 20,000 messages. Topology 1 consistently consumes less energy than Topologies 2, 3 and 4 but follows the same general trend of increased consumption with increased traffic volume.

By taking the average of the five trials at each traffic volume for each topology, we see that the outliers are eliminated and that the average number of nodes broadcast to falls between 25.6724 and 28.7712 for all of the topologies. For the traffic volume of 5,000 messages, the average number of nodes broadcast to and the anonymity factor is tightly grouped. At the traffic volume of 10,000 messages, the highest and lowest number average number of nodes broadcast to are both present. At 15,000 and 20,000 messages, the range that the average values are spread over decreases again. These results are shown in Table 1.

The results vary based on traffic volume and do not demonstrate any trends of convergence to a number of nodes broadcast to or divergence from a number of nodes broadcast to as traffic volume increases. Just as there are no trends in the average number of nodes broadcast to, there are no trends on the anonymity factor over the different traffic volumes. The anonymity factor is independent of the overall traffic volume is shown in Figure 4. This is an important conclusion because if the anonymity factor was reliant on a certain traffic volume this would be a constraint for the employment of the algorithm and our objective is for this to have broad applications.

Table 1. The results of the anonymity metrics is the average number of nodes broadcast to. This is used to determine the anonymity factor of the topologies.



| | Topology 1 | Topology 2 | Topology 3 | Topology 4 |
|---|---|---|---|---|
| Average Number of Total Number of Nodes Broadcast to by algorithm | | | | |
| 5000 Messages | 26.863320 | 26.743820 | 27.153320 | 27.100000 |
| 10000 Messages | 25.672740 | 28.771200 | 27.981800 | 28.036360 |
| 15000 Messages | 26.694700 | 28.320580 | 26.387500 | 27.237500 |
| 20000 Messages | 26.542860 | 28.451060 | 27.509640 | 28.314320 |
| Topology | 26.443405 | 28.071665 | 27.258065 | 27.672045 |
| Anonymity Factor | | | | |
| 5000 Messages | 0.037225 | 0.037392 | 0.036828 | 0.036900 |
| 10000 Messages | 0.038952 | 0.034757 | 0.035738 | 0.035668 |
| 15000 Messages | 0.037461 | 0.035310 | 0.037897 | 0.036714 |
| 20000 Messages | 0.037675 | 0.035148 | 0.036351 | 0.035318 |
| Topology | 0.037817 | 0.035623 | 0.036686 | 0.036138 |
| Average Anonymity Factor Across All Topologies | | | | 0.036566 |

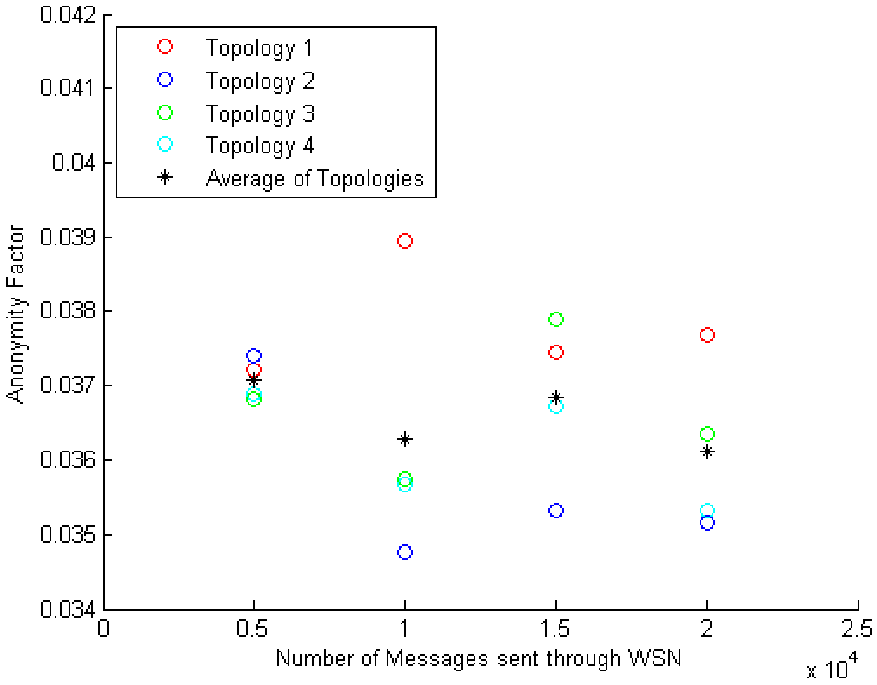


Figure 7. The anonymity factor of each topology and the average anonymity factor calculated at each traffic volume.

This conclusion leads us to examine the average number of nodes broadcast to at all message volumes, listed in Table 1, to determine the anonymity factor for the topology. We see very consistent results across the four topologies, as illustrated in Figure 5 and Figure 6. We see variation between the four topologies, just as we did in the energy efficiency conclusions. However, we also see that the results are remarkably consistent. The value of the anonymity factor for each topology is under 0.04. This means that for any given topology we simulated, an adversary conducting traffic analysis of the deployed WSN has a less than 4% chance of finding the sink node on his/her first guess when physically searching for the sensor.

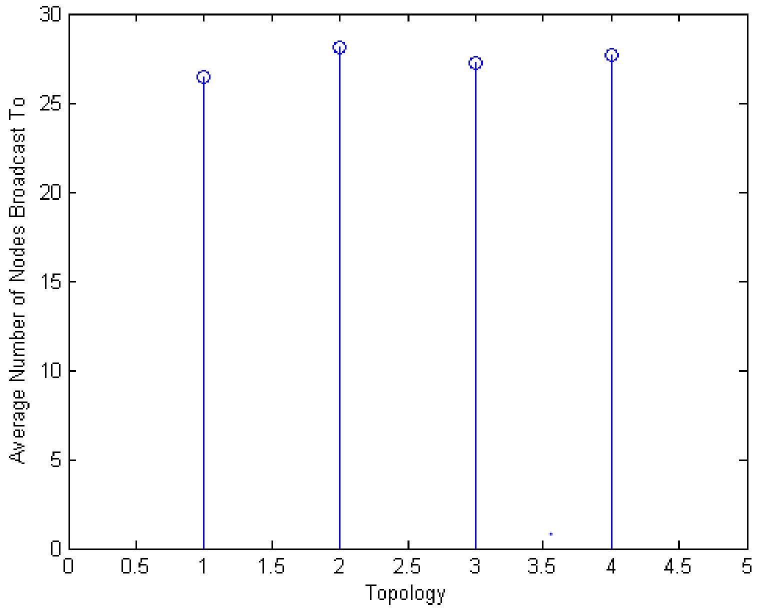


Figure 8. The average number of nodes broadcast to for each topology, with all traffic volumes included. After concluding that the anonymity is independent of traffic volume, we consider all of the data points for number of nodes broadcast to for each topology to determine the average.

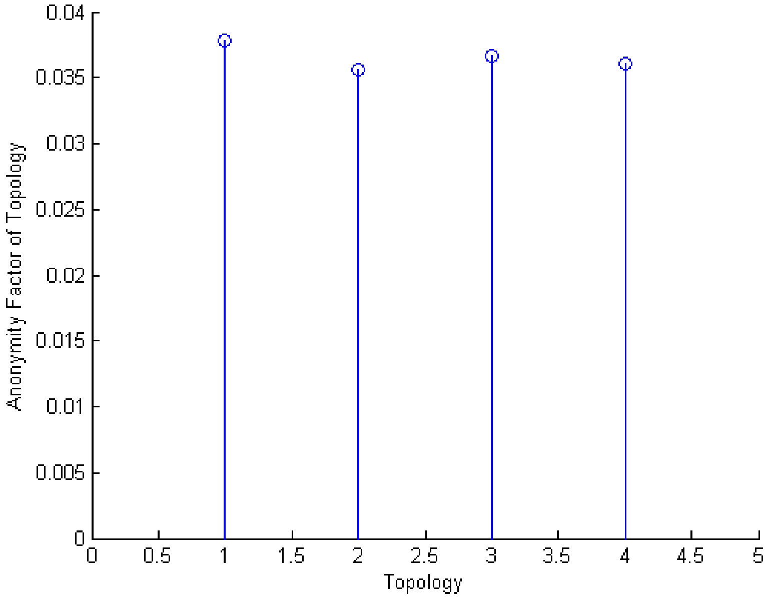


Figure 9. The anonymity factor for all four topologies. The anonymity factor of the topologies is calculated based on the average number of nodes broadcast to across all traffic volumes. The results are consistent across the topologies.

WSNs can be used for a variety of military, civilian and commercial applications. This thesis was motived by the proliferation of WSNs for military applications. The existing research focused on energy conservation without concern for WSN privacy or WSN privacy without concern for the limited resources of a WSN. The research in both areas failed to address realistic topologies for real world applications. We believe that bringing together the notion of energy efficiency and sink node privacy is vital to military applications of WSNs. The foundation to simultaneously achieve both objectives is provided by the results of this thesis.

## LIST OF REFERENCES

[1] M. Conti, “Body, personal and local ad hoc wireless networks,” in *The Handbook of Ad Hoc Wireless Networks,* M. Ilyas, Ed. Boca Raton, FL: CRC Press, 2003.

[2] K. Mehta, D. Liu and M. Wright, “Protecting location privacy in sensor networks against a global eavesdropper,” *IEEE Transactions on Mobile Computing*, vol. 11, no. 2, pp. 320–336, Feb. 2012.

[3] W. Stallings, “Data communications, data networks, and the Internet*,”* in *Data and Computer Communications*, 9th ed., Upper Saddle River, NJ: Prentice Hall, 2011.

THIS PAGE INTENTIONALLY LEFT BLANK

# ACKNOWLEDGMENTS

First and foremost, I would like to thank my husband, Gabe, for the support, patience, and love that has carried me through this process.

I would like to thank my thesis advisor, Professor Preetha Thulasiraman, for providing the guidance, direction, and support to complete this process.

To all of the professors in the Electrical Engineering Department who have been a part of my education, thank you for the knowledge.

THIS PAGE INTENTIONALLY LEFT BLANK

# I. INTRODUCTION

In 1997, the Institute of Electrical and Electronics Engineers (IEEE) adopted the first wireless local area network standard, named IEEE 802.11 [1]. The practical advantages of being able to move away from a wired architecture have driven staggering growth in the development of consumer and commercial devices that are able to connect wirelessly. Substantial improvements in integrated chips have also contributed to the miniaturization of devices, an increase in processing power resident in a device, and a rather dramatic reduction in cost per device.

Due to these technological advances, the manufacturing of small and low cost sensors has become technically and economically feasible [2]. A sensor observes an event or gathers some physical data from its area of interest. It then processes the observed or gathered data using a tiny embedded processor. The sensor sends the processed data to a central data collector either through direct wireless transmission or through intermediate nodes [3]. A basic sensor is composed of four subsystems: power, sensing, processing, and communications. The interaction of these four subsystems is illustrated in Figure 1. The power subsystem is a small battery with finite power capacity that is responsible for supporting the functions of all of the other subsystems. The capabilities of the sensing subsystem are very broad and can be tailored for desired applications. The sensing subsystem can be employed to gather meteorological variables such as temperature or pressure or for military use in surveillance missions to detect moving targets [3]. A small processor in the sensor comprises the processing subsystem. The processor is responsible for preparing sensed data for transmission. The communication subsystem is a Radio Frequency (RF) transceiver which is responsible for transmitting data from the sensor and receiving information from other sensors in the WSN. The Sensors may have additional optional subsystems, such as Global Positioning Systems (GPS) or mobilizers [2].

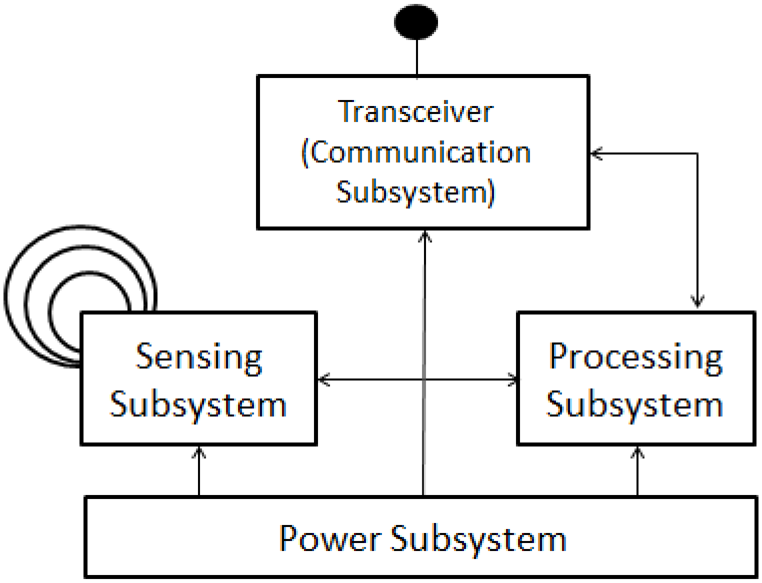


Figure 1. The basic architecture of a sensor consists of four subsystems including power, sensing, processing and communication subsystems, from [4].

## A. WIRELESS SENSOR NETWORKS: BACKGROUND

A wireless sensor network (WSN) is typically composed of a set of sensors that probe their physical environment for information and report their measurements to a nearby central controller. The controller aggregates all of the sensor node's information and interfaces the WSN to remote users who use the information to plan specific actions [5]. WSNs are ad-hoc networks in which sensor nodes are widely distributed in a region of interest for data extraction in real time. The sensor nodes act as both sensing and routing devices. Multiple sensor nodes may be used to transmit data from the initial source node to the destination (i.e., multi-hop communication). The destination node in a WSN is characterized as a sink node. A representative WSN topology for military applications is illustrated in Figure 2.

When a WSN is deployed, each sensor has a finite amount of energy. Sensors are powered by the power subsystem, and every action that is taken by a sensor has an energy cost that slowly depletes the sensor's power. Some actions like communication require a large amount of power, while other actions like processing and sensing data require a very small amount of power. When a sensor loses power, it is no longer able to sense information, communicate with other nodes or route information. The death of a single node does not have a major impact on the WSN, but as additional nodes die out, the performance of the WSN is degraded as the network may become partitioned and is no longer reliable. The tradeoff associated with small and inexpensive devices is that the network itself is resource constrained and has a limited lifetime.

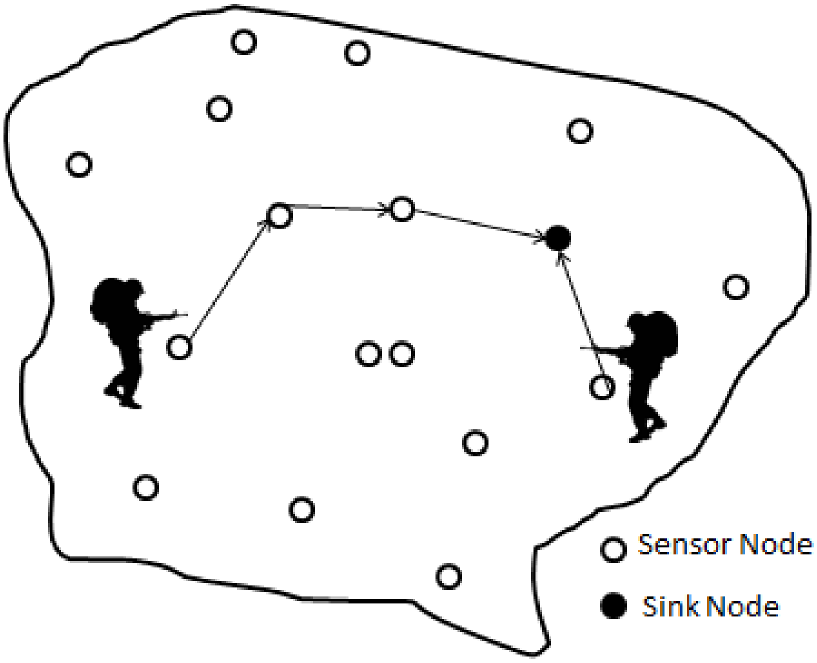


Figure 2. The basic topology of a WSN where sensor nodes are deployed to track the movement of personnel within an area of interest and report the sensed information back to the sink node, after [6].

## B. WSN APPLICABILITY TO THE DEPARTMENT OF DEFENSE

WSNs greatly extend our ability to monitor and control the physical environment from remote locations and improve the accuracy of information obtained via collaboration among sensor nodes and online information processing at these nodes [1]. For this reason, WSNs are currently used for a broad range of military, civilian, and commercial applications. Remote sensors provide a means to economically conduct continuous surveillance of vast areas, contributing key information to the intelligence collection effort. The Department of Defense (DOD) is able to make use of sensor technology to minimize risk to personnel during military operations and reduce the number of personnel required. Sensors can be placed just beyond the perimeter of a base, on and along the avenues of approach to provide early warning of incoming personnel and enhance perimeter security. In a disaster management setup, a large number of sensors can be dropped from the air and networked to assist in rescue operations and provide situational awareness [7].

WSN security is especially important from the DOD perspective. Remote Sensor Operations have long been a part of military operations. The Marine Corps began using sensor networks in 1967 during the Vietnam War [8]. These networks are a vital part of the United States Marine Corps (USMC) intelligence gathering efforts and expand the commander's view of the battlefield. Operational needs drive the intelligence gathering objectives. A WSN is deployed because additional intelligence is necessary to support the execution of an operational objective. The information collected by sensor nodes is distributed among small unit leaders for the planning and execution of tactical operations [8].

An important aspect of WSN security is the ability to protect the sink node. The sink node in a WSN is crucial for gathering, aggregating, and transferring sensor information. From the perspective of military applications, when sensors gather information, the central controller to which this data is sent is the sink node. Thus, the sink node is relied upon to provide critical information to personnel on the ground about an area of interest. Since the sink node is a central point of failure, an adversary can destroy the sink and render ineffective the data gathering duties of the entire sensor network. Thus, failure to protect the network completely subverts the intended purpose of sensor network applications [9]; therefore, it is important to implement specific protocols that conceal the sink node's location.

## C. RESEARCH MOTIVATIONS AND OBJECTIVE

Our study into WSNs is from a security perspective in that, because these networks are remotely deployed, they are vulnerable to malicious infiltration. The growing capabilities of WSNs and any potential adversary require some modification of the tactics, techniques and procedures used for the tactical employment of WSNs. It can no longer be assumed that an adversary has to be technologically advanced to observe or interfere with a deployed WSN. Due to the shared nature of wireless communication media, an attacker can easily eavesdrop on the radio communications either by purchasing their own sensor devices or by leveraging other radio devices capable of monitoring message transmission. Thus, no matter whether messages are encrypted or not, an adversary is able to identify contextual information [10]. While all traffic in a military wireless sensor network is encrypted, the contextual information that is revealed is meaningful-where the communication occurred and who participated in the communication. The role played by the sink node in the sensor network raises its profile as a high value target for attack; thus, sink node anonymity is crucial to the security of a wireless sensor network deployed for tactical use.

The privacy of the sink's location in a unique problem in WSNs. Most security and privacy research related to WSNs focuses on secure routing, key management, source privacy and denial of service. Nevertheless, the protection of the sink's location cannot be achieved using existing security mechanisms such as packet encryption, key management, etc. At the same time, a scheme for sink protection should not affect normal sensing and communication tasks that require knowledge of the sink's location. In most cases, sensed data is transmitted along paths from source nodes to a sink node. This produces pronounced traffic patterns that reveal the direction and, thus, the location of the sink node. An adversary can analyze the traffic patterns to deduce the location of the sink.

Due to the fact that traffic analysis is an effective mechanism to determine the geographic location of a sink, research concerning sink location privacy in a sensor network has attracted a lot of attention. By hiding the sink node's true location, the cost to the adversary to locate the sink node increases.

Another important parameter in achieving sink node anonymity/privacy is the issue of node energy maintenance. Any anonymity scheme that is implemented in a WSN must ensure that the energy of the nodes in not significantly depleted; thus, a balance must be achieved in which sink node anonymity is attained while keeping node energy levels sufficient enough to continue network operations.

## D. THESIS CONTRIBUTIONS

To address the issue of sink node privacy/anonymity, we develop a strategy to obfuscate the sink node's location using a hierarchical routing mechanism known as clustering while simultaneously limiting node energy depletion. To the best of our knowledge this is the first work that develops a novel sink node anonymity algorithm in a resource efficient manner. The contributions of this thesis are summarized as follows:

- Development and implementation of a network topology and clustering algorithm in a resource-efficient manner.
- Development of a routing algorithm for sink node anonymity.
- Simulation and evaluation of the routing algorithm for security robustness and energy preservation.

## E. THESIS ORGANIZATION

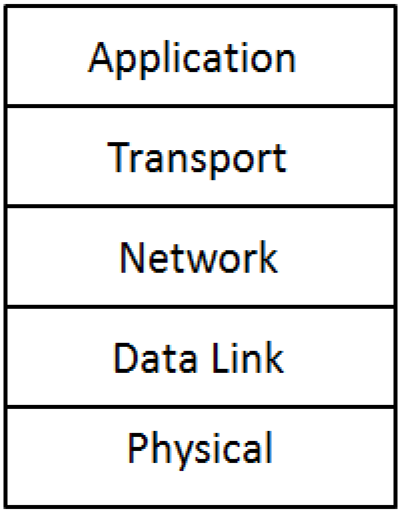
The remainder of this thesis is organized as follows. The current approaches to addressing the privacy and security of WSNs are outlined in Chapter II. The energy conservation schemes in WSNs are introduced in Chapter III. The basis of the experimental setup, the network model and threat model is explained in Chapter IV. The rationale behind the modeling decisions for the network parameters are also addressed in Chapter IV. The implementation of the clustering and anonymity routing algorithms using the network model developed in Chapter IV is discussed in Chapter V. The anonymity routing algorithm is simulated and evaluated in Chapter VI. The thesis is concluded and topics are proposed for future work in Chapter VII. All code for the algorithms implemented in this thesis is provided in the Appendix.

## F. CHAPTER SUMMARY

In this chapter we provide an introduction and overview of WSNs and their applicability to the DOD. The research motivations and objectives were discussed, followed by an outline of the thesis contributions.

# II. PRIVACY IN WIRELESS SENSOR NETWORKS

To defend and protect a WSN, it is necessary to understand the layering architecture of a network. A high degree of cooperation and coordination is needed for successful interactions between sensors. These interactions are complex and must be broken down into subtasks which are implemented separately [11]. The layering architecture of a network facilitates the implementation of these subtasks. The most common network layering model is based on the Open Systems Interconnection (OSI). The general network layering construct based off of the OSI model is shown in Figure 3. The architecture that defines the network functionality is split into layers that collectively form the protocol stack of the network [12]. Each layer in the stack performs a related subset of the functions required to communicate with another system. This protocol stack combines power and routing awareness, integrates data with networking protocols, communicates power efficiently through the wireless medium, and promotes cooperative efforts between sensor nodes [13]. Given this layered network architecture, we can analyze security issues at each layer and determine how security policies can be implemented at each layer.



Application

Transport

Network

Data Link

Physical

Figure 3. The five layers of the network on the OSI model, from [12].

The physical layer is responsible for frequency selection, carrier frequency generation, signal detection, modulation, and data encryption [13]. The most straightforward way to protect sensors is at the physical layer. Jamming and tampering are the major types of physical attacks. The standard defense against jamming involves various forms of frequency hopping communication which requires more complexity than low- power, low- cost sensors are able to employ. An attacker can tamper with nodes physically and interrogate or compromise them [14]. Passive tamper protection mechanisms including protective coating and tamper seals are common in sensors because they do not require additional circuitry or energy. While intrusion detection is an excellent first line of defense if the sensor is located, the basic functions are well known, and the implementation is left to commercial sensor manufacturers.

The data link layer is responsible for the multiplexing of data streams, data frame detection, medium access control (MAC) and error control. It ensures reliable point-to-point and point-to-multipoint connections in a communication network. The MAC protocol establishes communication links for data transfer [13]. The attacks at the data link layer compromise the availability of the WSN and deplete the battery of the nodes [14].

The transport layer provides end-to-end communication reliability for data exchanged by sensor nodes [13]. This layer is especially needed when the system will be accessed through the Internet or some other external network, as is the case for the sink node. The attacks that can threaten the security of the WSN at the transport layer are flooding and desynchronization attacks [14].

The application layer provides services for an application programs to communicate with the stack. For example, internet browsing uses an application protocol known as the hypertext transfer protocol (HTTP) to connect to the internet browser (the application) to the layering stack for the internet browsing experience.

There is abundant literature on privacy at the network layer. Privacy strategies implemented at the network layer require specific multi-hop routing protocols to be developed. These protocols are used to deliver data from a sensor node to the sink node while ensuring that privacy is protected [14]. There are a number of creative approaches to preserve WSN privacy at the network layer. They can be divided into two types: source-location privacy and sink- location privacy approaches [10].

## A. SOURCE NODE APPROACHES

The source node is the node where environmental sensing occurs. Failure to protect the source node's privacy can be detrimental for a number of reasons. As stated above, sensors are vulnerable at every level of the network protocol stack. If the security of a source node is compromised, it is open to detection, intrusion and interference. The military relies on WSN applications for intelligence collections [8]. If the privacy of the source node is compromised, then the adversary is able to locate and destroy a source node. Even without destroying the node, the adversary can undermine the WSN by shaping the traffic at the sensor node either by inflating the traffic volume or by deliberating bypassing the node. Data collection by sensors is a vital function of the network, and the compromise of a source node can subvert the utility of the WSN.

### 1. Periodic Collection

The simplest approach to protecting the source node is to require each source node in the network to transmit on a regular interval. The information transmitted can be sensed information or a dummy packet if the sensor has no information to relay at that time. An adversary observing the network will be unable to detect the location of the source node because the traffic patterns of the WSN are independent of the presence of real objects being sensed [9].

There are a number of limitations to this. Periodic collection can only be applied to applications that require data collection at a low rate and do not have a strict requirement for data delivery latency. This means that it is not practical for time sensitive WSNs. If the period of periodic collection is short, there is less latency in reporting real-time data. To implement short periodic collection times requires dummy traffic to be generated for each sensor which did not sense a real event during that period. The shorter the period, the more dummy traffic needs to be injected, shortening the network lifetime [9].

### 2. Source Simulation

In source simulation, a set of virtual objects are simulated within the WSN. The virtual object is similar to what would be sensed in a real event but is preloaded in a token, as the event does not physically occur. Each of the virtual objects generates a traffic pattern similar to that of a real object. Before the WSN is deployed, a subset of the sensors is pre-loaded with a token for the virtual object. Once deployed this token emits a signal used by real objects for event detection. This triggers the event detection process in the local area and generates traffic as if a real event is detected [15]. The generated traffic creates multiple viable traffic paths in comparison to the real traffic path, thereby obfuscating the source node. In Figure 4, one sensor node detects a real event, and the other sensor nodes with a token generate traffic that looks like a real event.

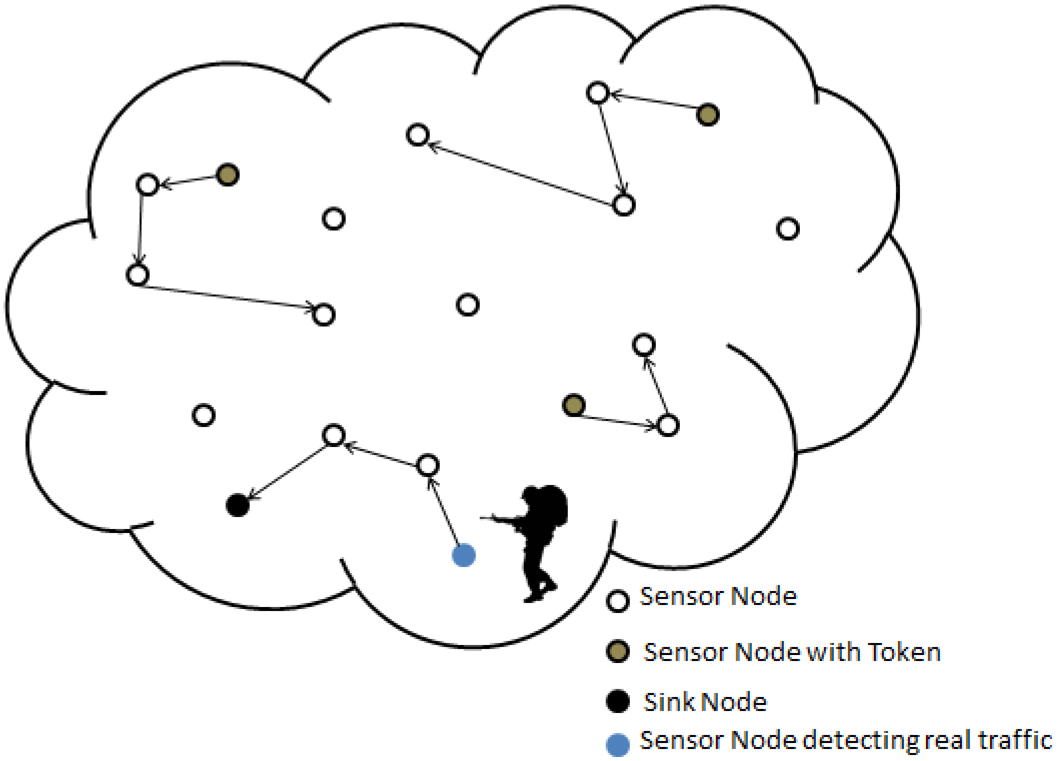


Figure 4. Virtual objects are simulated in the WSN. One path represents the movement of a soldier in the WSN as he passes sensor nodes; the other three paths are simulated alternates, after [15].

Source simulation applications are limited to mobile objects. The paths in Figure 4 represent a mobile object entering the sensing range of a number of sensors as it travels. Each sensor reports data back to the sink node. For example, the movement of a soldier on the battlefield can be picked up by multiple sensors as he conducts a patrol. The challenge presented by this is that the behavior of the mobile object needs to be predicted ahead of time so that it can be modeled within the network. An inaccurate model quickly reveals the fake traffic and real source node(s) [15].

### 3. FitProbRate

The FitProbRate scheme is a strategy introduced in [6]. The FitProbRate scheme allows for dummy traffic to be generated using an exponential probabilistic distribution function (PDF). FitProbRate improves upon the periodic collection approach discussed above by reducing the network transmission delay and overhead. The FitProbRate scheme is comprised of four algorithms which generate the traffic, calculate the delay to send real event data, and calculate the proper delay for subsequent dummy traffic to recover the mean of the probabilistic distribution. FitProbRate adjusts the flow of traffic to maintain the PDF and reduce real event latency [6]. This process is illustrated in Figure 5.

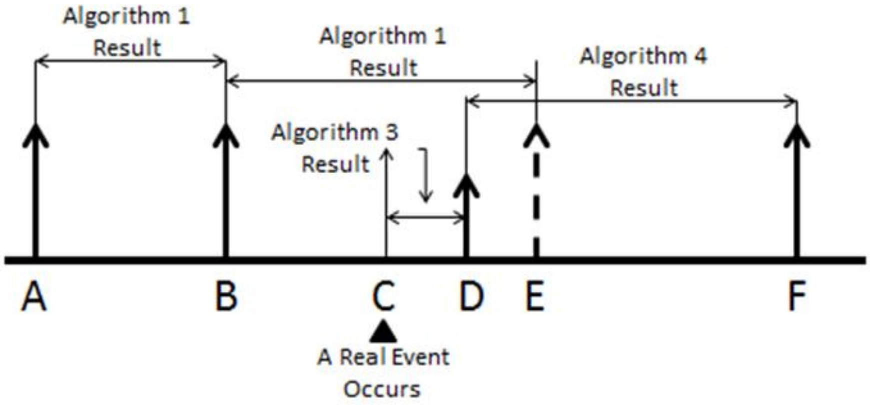


Figure 5. The example in this figure illustrates the entire FitProbRate process from determination of initial intervals and detection and transmission of real event data to adjusted transmission of dummy traffic to regain the mean of the PDF, from [6].

In Figure 5, A, B, and E represent the intervals during which dummy traffic is transmitted per the PDF. A real event occurs at C and is transmitted after a brief delay at D. The dummy traffic which would have been transmitted at E is instead transmitted at F to recover the mean of the PDF.

A disadvantage of FitProbRate is that the algorithms require significantly more computation than periodic collection. The advantage of this implementation is that there is a significant reduction in overhead associated with dummy traffic and data latency, making it suitable for a broader range of applications. The most significant limitation for FitProbRate is the scalability [6]. A large active network quickly returns to similar latency levels or has to generate significant dummy traffic, negating the advantages over simpler schemes.

## B. SINK NODE APPROACHES

The challenge of location privacy for the sink node is that the network traffic is asymmetric, with nodes further from the sink node seeing dramatically less traffic than nodes within immediate range of the sink node.

### 1. Deceptive Packets

Deceptive packets are generated from low traffic volume sensor nodes and take care to avoid routing through high traffic areas, ending their transmission at another low traffic volume node [5]. The deceptive packets protocol assumes that the adversary is conducting traffic analysis within the WSN and is able to correlate data transmissions to determine the end to end path. The Belief is a value which denotes the adversary's confidence that the destination node is the sink node [5]. The goal of using deceptive packets is to make the belief values of other nodes similar to or higher than the sink node. This approach is similar to the source simulation approach for source-location privacy. The two are differentiated by the method to generate these deceptive packets. Unlike source simulation where the nodes generating false traffic are seeded prior to deployment of the WSN, the deceptive packets protocol is adaptive. Sensor nodes use online data processing to evaluate the belief value for each node and determine where traffic should be generated from and where it is destined to go.

A disadvantage to the deceptive packet approach is that its performance is highly variable. In order to evaluate the belief values, the adversary must analyze the data it has collected. Deceptive packets utilize online processing to mimic the adversary’s belief calculations and determine where additional traffic should be generated. If the adversary is calculating the belief values at a different rate than the additional deceptive packets are being generated, then it is possible that the adversary may not be foiled by the deceptive packets. The largest limitation of this is that there is a significant amount of communication overhead associated with evaluating the belief and adjusting the volume and location of the deceptive packets. It is difficult to optimize minimizing communications overhead and normalizing the belief value of multiple nodes.

### 2. Location Privacy Routing

In the Location Privacy Routing (LPR) protocol, each sensor divides its neighbors into two lists: a closer list consisting of neighbors who are closer to the sink node, and a further list consisting of neighbors that are further from the sink node. When a sensor forwards a packet, it randomly selects a neighbor from one of the two lists. The route for multiple messages originating from the same source node is not always the same because the next hop is randomly selected. The two lists make it more difficult to predict the next hop and direction of the sink node because traffic does not always travel in the cardinal direction of the sink node [16]. Ultimately, this means that an adversary who is conducting a packet tracing attack has to take many more hops before reaching the sink because it is frequently deviated in the wrong direction.

If we apply LPR alone, the protection for location privacy is not significantly strong. This is because the overall traffic trend in the network still points towards the sink node. Although this problem can be alleviated by increasing the probability that a sensor forwards to a neighbor on the further list, it leads to a longer delay and higher energy costs [16].

One way to overcome this is to combine LPR with fake packet injection similar to deceptive packets. The basic idea of fake packet injection is that when a sensor node forwards a real data packet, it may generate a fake packet and transmit it to a neighbor randomly chosen from the further list. This leads an adversary away from the sink node, distributes the direction of outgoing packets while reducing data latency for real data, and increases the location privacy of the sink node in the WSN. These methods complement one another but are ultimately challenged by a global adversary who can see that all real messages ultimately always arrive at the sink while fake messages do not.

### 3. *k*- anonymity

The goal of the *k*-anonymity algorithm is that at least *k* entities exhibit the same characteristics as nodes located close to the sink. In order to achieve *k*-anonymity, a Euclidian minimum-spanning tree-based routing algorithm is proposed to route traffic so that traffic volumes are equally high at *k* sensor nodes in the WSN. Since at least *k* nodes exhibit similar traffic statistics, an adversary trying to locate the sink node has to locate and inspect all nodes within the communication range of each node [10].

However, positioning *k* designated nodes within the WSN is complex as it affects two conflicting goals: the routing energy cost and the achievable privacy level [10]. This is ultimately an optimization problem which requires prioritizing one goal or the other.

### 4. Randomized Routing with Hidden Address

The methods discussed thus far have assumed a passive adversary whose methods are limited to observing network traffic. An active attacker can compromise a node and read the header field of a packet to identify the receiver.  The Randomized Routing with Hidden Address (RRHA) scheme keeps the identity of the location of the sink secret in the network. Sensors do not know who and where the sink is when routing packets and do not specify a destination when reporting their measurements. The packets are forwarded along different random paths for a specified path length and are then discarded when the length is reached [17].

The random path taken by RRHA introduces some packet delay. The longer a packet lingers in the WSN, the more energy it consumes. When there is high traffic volume, the delay caused by the random paths can accumulate to cause significant network congestion, exaggerating the delay further and degrading the performance. The major limitation of RRHA is that it cannot guarantee that the sink will receive the data. Simulations showed that the longer the path length, the higher the success rate of information reaching the sink [17]; however, in many time sensitive applications this is clearly an unsatisfactory outcome.

## C. CHAPTER SUMMARY

We know that the most successful protection for a sink node against a malicious adversary's attack is to remain anonymous in role, identity, and location [18]. Detailed within this chapter is some of the current research on how to address privacy in wireless sensor networks. It is important to note the limitations associated with these approaches as they are the compelling reason to continue research in this field.

THIS PAGE INTENTIONALLY LEFT BLANK

# III. ENERGY CONSERVATION IN WIRELESS SENSOR NETWORKS

Energy conservation in a WSN is a crucial issue as sensor nodes are all powered by limited battery sources. Energy efficient design of a WSN has drawn considerable attention from many researchers. This has resulted in the development of various approaches for saving the limited energy of the sensor nodes, thereby extending the life of the network [19, 20, 21, 22].

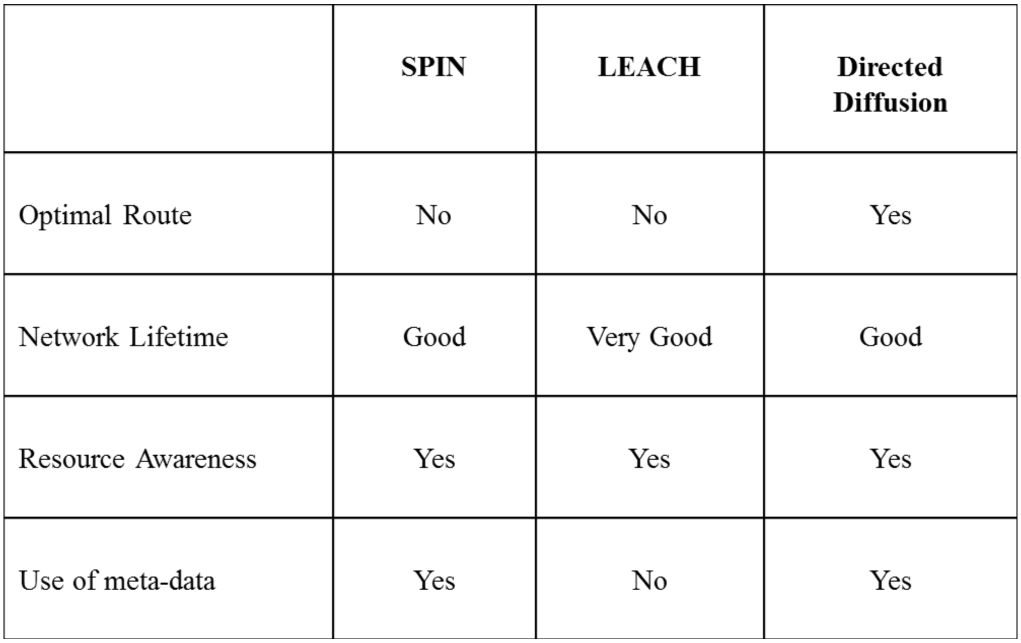
Sensors utilize their energy for sensing and processing data as well as transmitting and receiving data. The communication subsystem of a sensor node (as discussed in Chapter I, Section A) consumes more energy than the processing subsystem. It has been shown that transmitting one bit of data may consume as much energy as executing a few thousand computational instructions [19]; thus, it is important that energy efficiency be targeted towards the communications subsystem as only minimal gains are attained by optimizing the energy of the sensing and processing subsystems. In order to develop energy efficient communication mechanisms in a WSN, we focus on the network layer of the protocol stack. Efficient algorithms can be developed at the network layer such that reliable route setup and relaying of data from the sensor nodes to the sink is achieved and the lifetime of the network is maximized [7].

## A. ENERGY EFFICIENT ROUTING

Energy efficient routing protocols for WSN can be broken into three broad categories: data centric protocols, hierarchical or clustering based protocols, and location based or geographical protocols. Within these categories, three popular approaches have emerged. Data centric routing techniques utilize a query driven model to reduce the amount of transmitted data and are also able to aggregate data while relaying it to the sink. Directed Diffusion and Adaptive Protocols for Information Dissemination in Wireless Sensor Networks (SPIN) are the dominant data centric protocols [3]. Low Energy Adaptive Clustering Hierarchy (LEACH) is chief among the hierarchical or clustering based protocols.

These three methods have become common performance baselines, with the majority of energy efficient routing research focused on improving their performance. Some of the parameters which are used to evaluate these routing protocols are compared in Table 1.

Table 1. SPIN, LEACH, and Directed Diffusion are compared with one another, from [2].



| | **SPIN** | **LEACH** | **Directed Diffusion** |
|---|---|---|---|
| Optimal Route | No | No | Yes |
| Network Lifetime | Good | Very Good | Good |
| Resource Awareness | Yes | Yes | Yes |
| Use of meta-data | Yes | No | Yes |

### 1. Directed Diffusion

Directed diffusion finds routes from multiple sources to a single destination that allows in-network consolidation of redundant data (aggregation) [20]. The sink node advertises an *interest* or what information it is interested in receiving. The interest is propagated through the WSN. Each node that receives the interest remembers who sent it and sets up a gradient, which is a list of neighboring sensors which have the same interest. Upon sensing an event matching the sink node's interest, the sensor generates a data packet and sends it to the sink via the neighbors for which it has a gradient. A node that receives this message checks if it has received the identical message before. If an identical data item exists in the cache, the node drops the message. If this data item does not exist in its cache, the node determines the matching interest and resends the data along the gradient towards the neighbor [20]. This process is repeated by each node receiving the data packet. The data packets are called exploratory packets since they are sent to the sink along multiple paths. Eventually, in the last phase of directed diffusion, the sink reinforces the path from which, for example, it received the first exploratory data packet. This means that only one path is selected from the sink to the packet source, and this is the route that is used by the sensor to deliver data to the sink. The general approach that directed diffusion takes is illustrated in Figure 6.

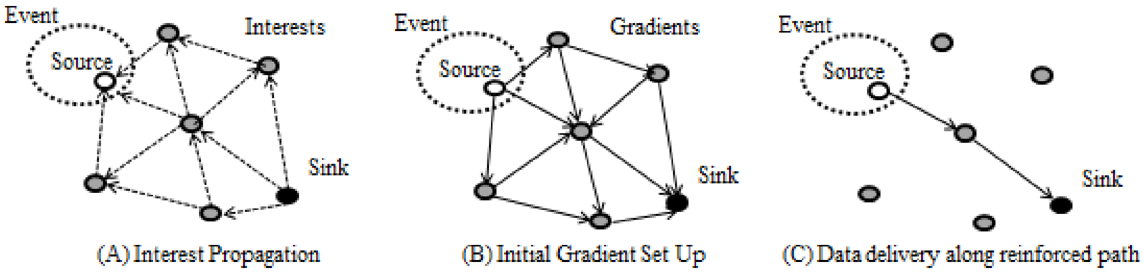


Figure 6. A simplified schematic for directed diffusion, from [21].

Directed diffusion has several disadvantages which limit its application. The query driven data delivery in this algorithm is a potential liability because many WSNs require continuous or periodic data delivery to the sink node. This requires the sink to constantly be requesting information from the source nodes, reducing the proposed energy savings and increasing congestion on the network. Also, the gradients which are selected to route traffic from the source to sink do not perform any load distribution. The weakness in this approach is that if one sensor or group of sensors is particularly active, they could deplete the energy on one path to the extent it is rendered unusable even if the lifetime of the WSN is extended. These are two limitations that can potentially be improved upon with minor modification to the directed diffusion protocol [21]. The largest disadvantage of the directed diffusion algorithm is the lack of location privacy it provides. In particular, traffic using directed diffusion converges towards the sink node, ultimately revealing the sink node’s identity as well as the identity of nearby high volume sensor nodes. There is little that can be done to mitigate this within the directed diffusion framework, leaving the network vulnerable to high value target attacks.

### 2. Sensor Protocols for Information via Negotiation

SPIN relies on two key innovations to manage the energy consumption of sensors in a WSN: negotiation and resource- adaptation. [22]. SPIN nodes negotiate with each other before transmitting data. Negotiation helps ensure that only useful information is transferred. Meta-data is exchanged in SPIN negotiations to describe the information a node has to share and prevents the flooding of redundant data messages through the WSN. Each sensor node has its own resource manager that keeps track of resource consumption; applications probe the manager before transmitting or processing data. This allows sensors to cut back on certain activities when energy is low including forwarding third party data.  This process is illustrated in Figure 7.

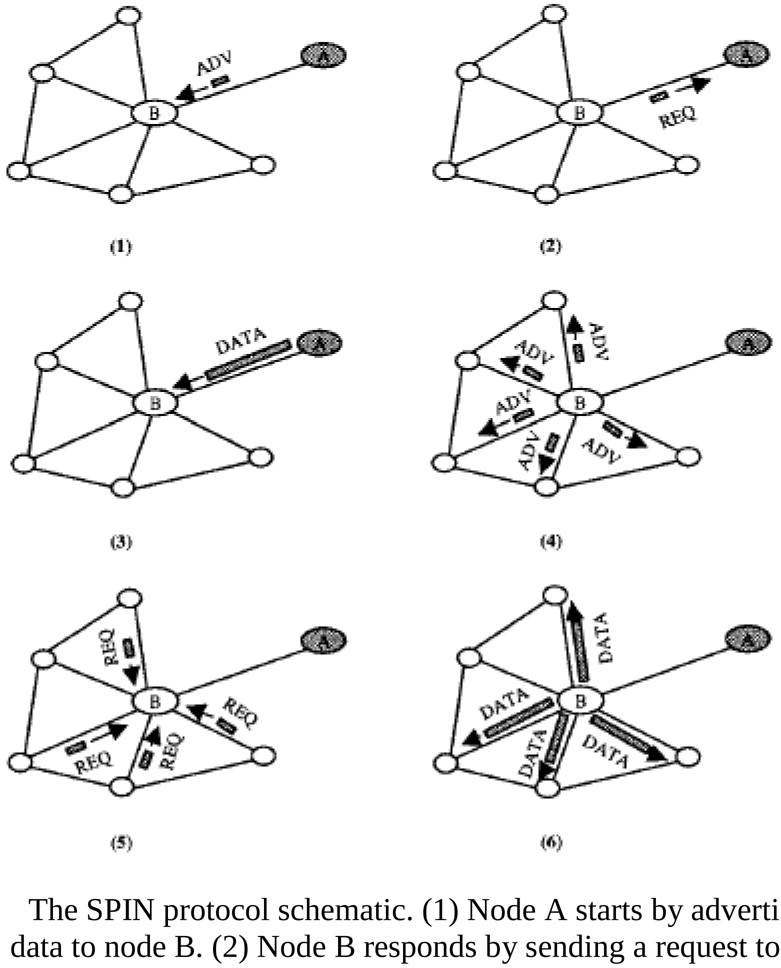


Figure 7. The SPIN protocol schematic. (1) Node A starts by advertising its data to node B. (2) Node B responds by sending a request to node A. (3) The requested data is received. (4) Node B then sends out advertisements to its neighbors. (5) Neighbors respond by sending a request back to node B. (6) Node B sends the data to neighbors who requested it, from [22].

SPIN offers an improvement on flooding, but there are still limitations and inefficiencies. One challenge with SPIN is related to the meta-data descriptors which advertise the data. To achieve energy savings, these advertisements must be smaller than the data itself but must be unique and descriptive enough to inform the neighboring nodes of the available information. The largest limitation of SPIN is that SPIN's data advertisement mechanism cannot guarantee delivery of data [21]. If intermediate nodes between the source and sink are not interested in the data which is being advertised, then the data is not delivered to the destination.

Improvements to SPIN which address some of these liabilities have been proposed in MSPIN [23] and SPIN-1 [24].

### 3. Low Energy Adaptive Clustering Hierarchy

Clustering transformed a flat network into multiple tiers, known as clusters, to segments the network. The partition of the WSN into clusters is illustrated in Figure 8 as a Voronoi diagram.

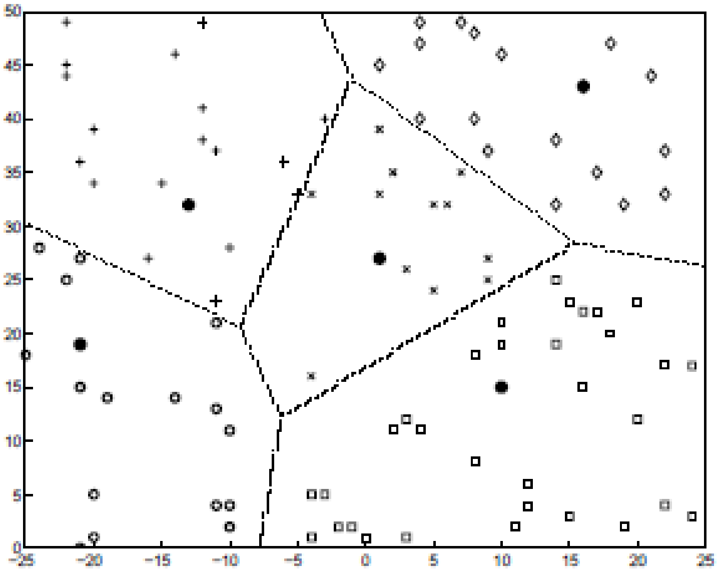


Figure 8. Representation of a WSN segmented into clusters with each cluster having its own CH. Cluster heads are represented by the solid circles in the diagram and sensor nodes are represented by open circles, from [25].

Cluster heads (CH) are used to perform data aggregation and/or data fusion before forwarding information onto the sink. Low Energy Adaptive Clustering Hierarchy (LEACH) is a clustering based protocol that aims to minimize energy dissipation in sensor networks [25]. Sensor nodes form clusters and elect CHs which are then responsible for transmitting data to the sink node. Nodes within the cluster achieve energy savings by transmitting only to the CH. LEACH then rotate CHs to distribute energy requirements among all the sensors. Additionally, LEACH performs local computation at each CH (data aggregation) to reduce the amount of data that must be transmitted to the sink. This saves both energy and bandwidth.

LEACH was originally developed when sensor technology was quite new. Thus, there are a number of limitations to its practical application for current situations. LEACH assumes all nodes can transmit with enough power to reach the sink if needed, which limits its utility for a WSN deployed over a large area [2]. In this sense, LEACH is not scalable for a broad number of applications. Also limiting the application of LEACH is that it was developed for sensing at a fixed rate and cannot support event driven or time sensitive reporting. The biggest limitation of LEACH stems from the fact that its primary focus of LEACH is of the network lifetime. It was not developed with security as a concern and has no features which address the security or privacy of data within a WSN.

In the years since LEACH was published there has been additional research to address some of these limitations including E-LEACH, M-LEACH, LEACH-C and V-LEACH [3]. However, the solutions proposed in these LEACH extensions are not comprehensive.

## B. CHAPTER SUMMARY

The current models and approaches to energy conservation in a WSN were detailed in this chapter. We discussed various energy saving mechanisms introduced in the literature and their limitations in terms of network security so that we can best evaluate the performance gains of our research.

# IV. EXPERIMENTAL SETUP

Modeling a WSN to use in an experimental setup is particularly challenging because different applications require different modeling parameters. In this experimental setup, we strive to make the model as realistic as possible within the confines of the simulation platform. While privacy concerns of WSNs have been addressed in the literature, many do not use a realistic model that mirrors real world applications of WSNs, thus limiting their practicality. Our research motivations stem from the use of WSNs for military applications; therefore, it is our intention to make modeling decisions in line with real world use.

## A. NETWORK MODEL

Commercially available and military grade sensors come with a wide range of capabilities. In this section we discuss these variations and the assumptions we make for our network model.

### 1. Sensing Range and Transmission Range

The sensing range of a sensor is the maximum distance which a sensor can sense. More specifically, any event is said to be detectable if it lies within the sensing range. The transmission range of a sensor is the maximum distance which a sensor can communicate information. In this thesis, the sensing and transmission ranges for all the sensor nodes are uniform. The sensing and transmission ranges are controlled by two different subsystems of the sensor node and do not have to be equal. In this thesis we assume that the sensors are placed within sensing range of their target, and we are not concerned with the difference between the sensing and transmission range. We also assume that each sensor node is equipped with an omni-directional antenna which allows the sensor to sense and communicate in every direction, and the transmission range is fixed at 40 meters. A sensor is able to exchange information with all neighboring nodes within this range.

### 2. Event Driven and Periodic Reporting

WSNs generally fall into one of two categories for sensing and communicating information. In event driven reporting, the sensors of a WSN immediately relay the information that they sense. In periodic reporting, the sensors of a WSN collect information and relay it on a fixed schedule. For military use, we assume all information is relevant and time sensitive; thus, we assume event driven reporting.

### 3. Sink Node Resources

It is assumed that all of the nodes have identical resources with the exception of the sink node. The sink node has the same fixed transmission range as the other nodes but has more processing and power resources to handle traffic volume and relaying of information outside of the WSN.

### 4. Data Exfiltration from the Network

The sink node is assumed to be inside “friendly lines” and is not sensing but only receiving traffic from the network.  The sink node acts as a gateway between the multi-hop network of sensor nodes and the wired network infrastructure or a repository where the sensed information is analyzed [10]. We assume that once this information arrives at the wired network, it is not vulnerable to malicious traffic analysis and does not have the same privacy concerns.

### 5. Number of Sink Nodes

WSNs may have more than one sink node. For simplicity, we assume that the network only has one. Additional sink nodes can be utilized for load balancing or redundancy to increase network reliability. While the contributions of this work can be expanded to show similar results with multiple sink nodes, we do not investigate the multi-sink scenario in this thesis.

### 6. Passive Receipt of Messages

There is no acknowledgement of messages received in the WSN. This means that the transmitting sensor node has no way of knowing whether or not the information it reports arrives at the sink node. This is standard practice in WSNs as the tradeoff is made between a two-fold increase in network traffic and conservation of limited network resources.

### 7. End to End Encryption

It is assumed that information is encrypted at the sensing node and is not decrypted until it reaches the sink node. It could be decrypted and encrypted at each hop but would require more processing power from each intermediate node.

## B. THREAT MODEL

The sink node is the aggregating point for data collection within the WSN. This is a high value target for the enemy. By locating the sink's physical location an enemy can attack it and, thereby, affect a commander's ability to utilize the WSN to collect battlefield intelligence and plan operations. The capabilities of the adversary affect how we choose to defend the network and evaluate the success of the proposed algorithm. We assume the following capabilities.

### 1. Global Knowledge

An attacker can easily eavesdrop on the radio communication either by purchasing his/her own sensor devices or by leveraging other radio devices capable of monitoring message transmission [11]. In doing so the attacker is able to view all traffic on the WSN.

### 2. Passive Observation

The adversary is not interested in interfering with the regular communications of the WSN. Attackers will use information like packet transmission time and frequency to perform traffic analysis and infer the locations of monitored objects and sinks [10].

### 3. Encryption

An advanced adversary will be aware of the encryption protocols utilized; however, we assume that he/she will not possess the encryption key. The adversary will then only able to ascertain contextual information such as traffic volume, the number of messages that arrive and depart each node, and possibly the cardinal direction of the traffic.

## C. THE MATLAB MODEL

The MATLAB model used in this thesis is constructed based on the following parameters.

### 1. Simulation Setup

The model which will be used in this experiment is a square 100 meter by 100 meter area. There are 100 nodes in the model. This number can easily be adjusted but is a reasonable number based on the size of the geographic area and range of nodes. One hundred nodes ensure ample coverage of the area of interest and connectivity of the WSN.

### 2. Placement of Nodes

The nodes are randomly distributed throughout the entire area. Nodes can be placed one of three ways: air, mounted patrol, or foot patrol [9]. In the case of the aerial and mounted patrol emplacement of the sensors, the distribution can best be described as random. On a foot patrol the placement of the sensor nodes is more careful and deliberate in the area of interest. From a global view of the sensor area, the placements do not follow any pattern and can be modeled by a random distribution as well. This model applies to all three placement methods.

### 3. Placement of Sink Node

The sink node is deliberately placed at the location $(x,y)$=(25 meters, 75 meters). The location of the sink node is deliberate because the team responsible for deploying the WSN deliberately places the sink node, likely co-locating it with their observation post. The exact coordinates are not significant in this experimental setup. The only significance is that the sink node is not randomly placed.

All of this information is coded in MATLAB, and the final result is depicted in Figure 9. The MATLAB code is included in an appendix at the end of this thesis.

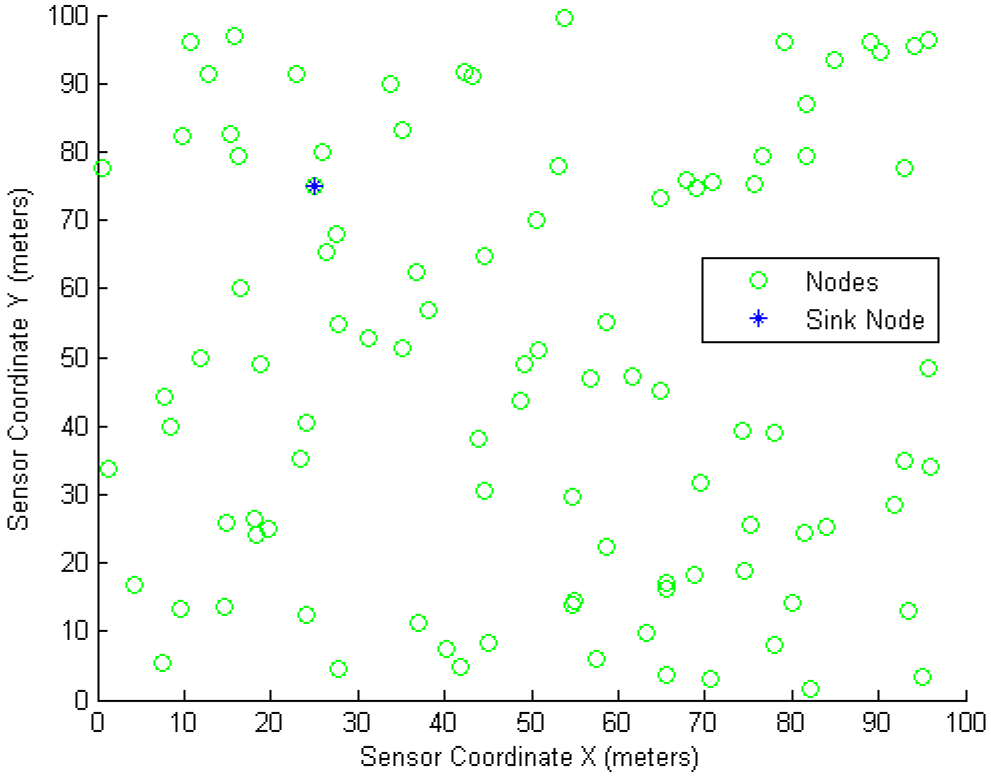


Figure 9. The MATLAB model of the experimental setup is 100 m × 100 m with 100 nodes.

## D. CHAPTER SUMMARY

In this chapter we explained the experimental setup and modeling assumptions. The MATLAB model of the physical topology which simulations were run on was also introduced.

THIS PAGE INTENTIONALLY LEFT BLANK

# V. CLUSTER BASED ROUTING TO ACHIEVE ANONYMITY

As discussed in Chapters II and III, there is a substantial amount of ongoing research in the fields of both privacy and energy conservation in WSNs. In order to achieve energy constrained anonymity, we propose a routing algorithm based on node clustering which results in at least *n* other nodes having similar observable traffic statistics, thus obfuscating the sink node's location.

The steps that the WSN takes upon deployment to route traffic are as follows:

- CH election and cluster formation. The election of CHs is discussed in Section A1 of this chapter.
- Choose a subset of the CHs to serve as broadcast CHs. The election of broadcast CHs, their importance in the network and the role they play in achieving sink node anonymity is discussed in Section A3 of this chapter.
- CHs use Dijkstra's algorithm to determine their route to the sink node's CH. Dijkstra's algorithm is discussed in Section B of this chapter.

## A. CLUSTERING

Clustering is a standard approach for achieving efficient and scalable performance in sensor networks. Clustering nodes into groups saves energy and facilitates distribution of control over the network [25]. To form clusters, sensor nodes must first elect a CH for each cluster. Nodes in the WSN which are not CHs find the closest CH within range and become cluster members. The nodes in a cluster only communicate with one another and the CH. Data sensed by a node is transmitted to its CH. The CH is responsible for all routing and communication external to the cluster. This yields energy savings over a "flat" topology, where each node must determine the route from source to sink node. For these reasons, the first step in our proposed algorithm is the initialization and formation of clusters. All of the nodes in the WSN either elect to become a CH or join a cluster as a cluster member, with the exception of the sink node. The sink node is always a cluster member in the WSN; it is never elected to be a CH. We force this constraint on the sink node because, if the sink node is always a CH, then it becomes clear to an adversary conducting traffic analysis that after a few CH rotations the sink node is the only node constantly re-elected to the role of CH. This leads the adversary to conclude the sink node (one of several CHs) has a more significant role in the WSN.

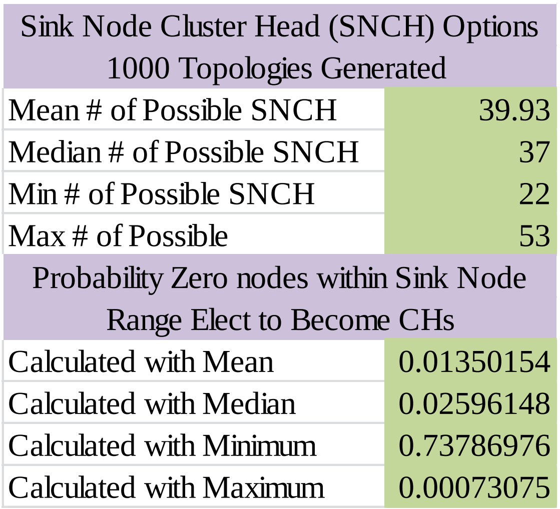
### 1. Cluster Head Election

Each sensor in the WSN may elect to become a CH with a fixed probability *p* when the network is deployed. There is not an optimal number of CHs for a WSN. For every topology the clustering process must ensure that no nodes become isolated and that there are no more clusters than necessary as excess clusters reduce the energy savings yielded from clustering.

An iterative approach is utilized to balance the competing demands of preventing isolation and achieving energy efficiency. In this thesis the probability of a sensor node electing to become a CH *p* is fixed at 0.20. This value was experimentally determined so that most of the CHs are elected in the first iteration, while the additional two iterations serve to ensure that no sensor node is isolated in the WSN.

As stated earlier in this section, the sink node is never a CH; therefore, the sink node does not go through the process of electing to become a CH. The sink node simply looks for the nearest CH to join as a cluster member. The CH that serves the sink node is referred to as the sink node's CH. We determined over 1000 different topologies where the mean number of possible sink node CHs is 39, with the minimum being 22 and the maximum being 53, so there is always be a node within range to serve as the sink node's CH. If each of these nodes elects with a probability of 0.2 to become a CH, then there is a 0.01% chance (based on the average number of nodes) that none of these nodes elect to become a CH. If this condition happens, then the network reinitializes and repeats the CH election process. These simulations and calculations are detailed in Table 2, where SNCH refers to the sink node's CH.

Table 2. One thousand different topologies were generated and evaluated to determine the probability that the sink node does not have a CH within range.



| Sink Node Cluster Head (SNCH) Options<br>1000 Topologies Generated | |
|---|---|
| Mean # of Possible SNCH | 39.93 |
| Median # of Possible SNCH | 37 |
| Min # of Possible SNCH | 22 |
| Max # of Possible | 53 |
| Probability Zero nodes within Sink Node<br>Range Elect to Become CHs | |
| Calculated with Mean | 0.01350154 |
| Calculated with Median | 0.02596148 |
| Calculated with Minimum | 0.73786976 |
| Calculated with Maximum | 0.00073075 |

In the MATLAB model, each iteration of CH election is marked with a different symbol; however, they all fulfill the same role. Using different symbols facilitates tracking and visualization of the steps of the iterations. There is no requirement that each iteration yield additional CHs. For this thesis we model a 100 node WSN; however, for clarity and simplicity, the images in this chapter are for a 30 node WSN.

### *a. First Iteration*

In the first iteration, each node may elect to become a CH with a probability *p*. If a node does not become a CH, then it determines if there is a CH within transmission range. At the end of the first iteration, nodes belong to one of three categories: 1) node is a CH, 2) node is within range of a CH and 3) node is not a CH or within range of a CH. The first iteration of CH election is shown in Figure 10.

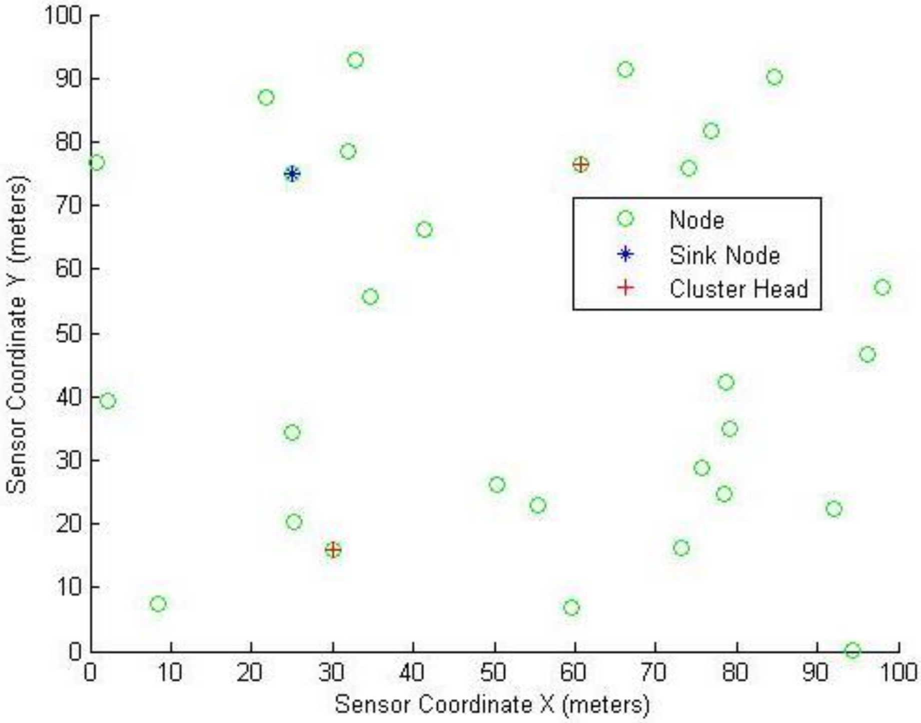


Figure 10. The first iteration of CH elections where the CHs are denoted by a red plus sign and the sink node is denoted by a blue star. Open green circles represent the remaining sensor nodes.

### *b. Second Iteration*

In the second iteration, all nodes that belong to category three at the end of the first iteration again elect to become a CH with probability *p*. All nodes which have not elected to become a CH in either iteration find the nearest CH within transmission range and elect to become a cluster member. The second round of CH election is shown in Figure 11. If desired, the steps of the second iteration can be repeated as additional iterative steps. The benefit of additional iterations is that a lower initial *p* can be used. Using a lower *p* results in a more gradual election of additional CHs. With each iteration a few more CHs are elected until there is adequate connectivity coverage across the WSN. The more gradually CHs are elected, the more optimal the final number of CHs; however, there is an energy cost associated with executing each iteration. In this thesis the total number of iterations is kept to three. We found that three iterations are sufficient to ensure that no nodes are isolated, and all nodes belong to a cluster.

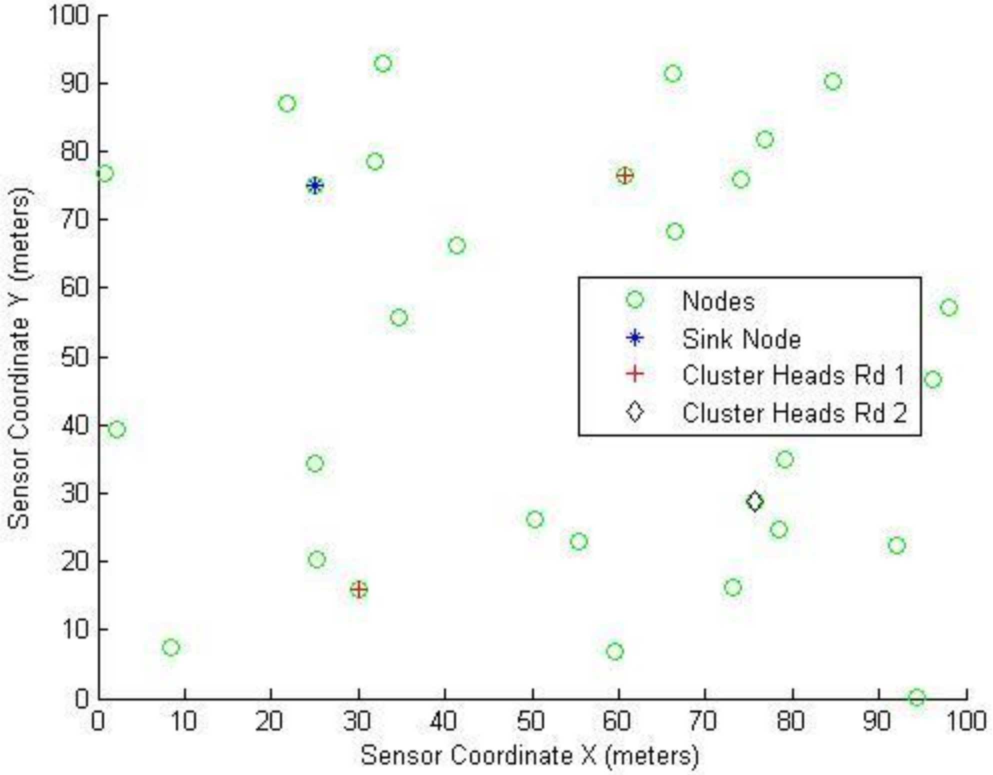


Figure 11. The second iteration of CH elections. The newest CHs elected during this iteration are denoted by a black diamond.

### *c. Final Iteration*

In the final iteration, any remaining nodes which are not in a cluster, that is not a CH or a cluster member, elect to become a CH. The final representation of the WSN with all elected CHs is shown in Figure 12.

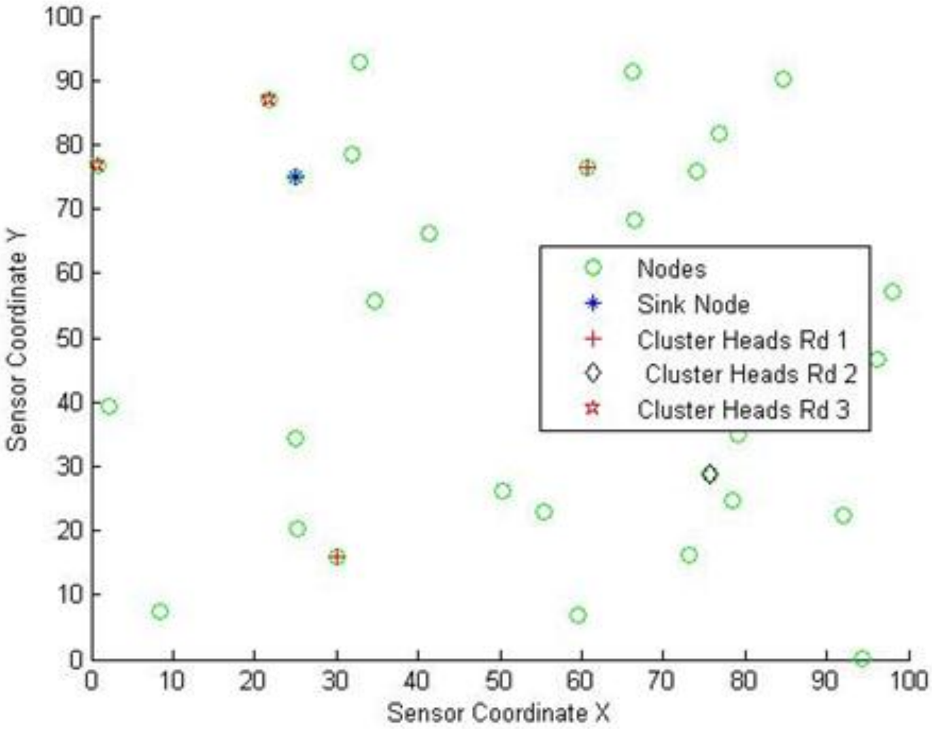


Figure 12. The final round of CH elections, where all nodes which are not part of a cluster become CHs. These CHs are denoted by a red star.

### 2. Rotating the Cluster Heads in the WSN topology

The use of a clustering hierarchy improves the overall energy efficiency of the WSN. Only CHs calculate routes and route traffic, which is a considerable savings over each node acting independently to route traffic. Clustering also imposes a substantial energy burden on the nodes that act as CHs; therefore, it is necessary to rotate the role of CH within the WSN.

We rotate the CHs for two reasons: load balancing and privacy. The CHs are reelected in the same manner they were initially elected. The CHs are rotated when one of two conditions are met. Either one of the CHs has expended a certain amount of energy or a specific number of messages have been transmitted through the WSN. Implementing CH rotation allows us to distribute the burden of being the CH across the WSN while increasing the overall lifetime of the WSN. The CHs are rotated if 1) any CH expends one percent of its initial energy value $E_o / 100$, where $E_o$ denotes the initial node energy or 2) the sink node's CH receives 1000 messages. We set the energy threshold to one percent because the energy costs of routing traffic in the WSN are relatively low. If we waited for more energy to be consumed, for example 5%, the CHs would only rotate when the number of messages threshold was met. We choose to rotate fairly often because we do not want the cluster topology of the WSN to be static for long periods of time. With a static topology it is plausible that the adversary could locate and inspect each node for which traffic is broadcast to in an effort to find the sink node [10]. Rotating the CHs increases the privacy of the sink node by randomizing the paths that traffic takes through the WSN and makes it more difficult for an adversary to draw any conclusions as to the location of the sink node.

### 3. Broadcast Cluster Head Election

The CHs in the WSN are responsible for routing data from the source node's CH to the sink node's CH. When forwarding data to the next node, each CH has two options. The message can be directly forwarded to the next node or widely broadcast to all sensors within range. In this algorithm we propose that a subset of CHs is selected to broadcast. One key consideration to broadcasting is overhead. We are aware that information is being transmitted to nodes that do not need it. In order to reduce overhead and limit broadcast information, we only allow a subset of CHs to broadcast to their members. The sink node's CH always broadcasts the messages it receives so that the sink node can receive the information. By broadcasting traffic to nodes other than the sink, we are essentially creating a situation where multiple nodes resemble the sink in terms of traffic volume. In other words, from the adversary's perspective, these multiple nodes are acting like sink nodes. In addition to the traffic volume, the cardinal direction of traffic is also disturbed. An attacker cannot use traffic volume for traffic direction to determine a sink node's location; thus, the cost of attacking each of these nodes is much higher than attacking just one (the sink node).

In choosing the broadcast CHs there are two key considerations: 1) The amount of residual energy remaining for the CH and 2) the number of cluster members of each cluster. The total number of broadcast cluster nodes is variable based on the number of members in each node. A lower threshold of 20 nodes broadcast to is established in this algorithm to ensure a minimum desired level of anonymity. The number of nodes broadcast to directly correlates to the anonymity of the sink node, as discussed later in this Chapter in Section D.

The CHs are ordered by their residual energy levels. The CH with the most energy is chosen to be the first broadcast CH. The number of cluster members which are broadcast to is then saved. Each subsequent broadcast CH is selected sequentially based on the most residual energy. The number of cluster members broadcast to is added to the previous value and, when the lower threshold for the number of nodes broadcast to (i.e., 20) is exceeded, no additional broadcast CHs are selected.

## B. DIJKSTRA'S ROUTING ALGORITHM

Once broadcast CHs are determined, we must determine the paths that traffic takes to reach the sink node's CH. Note that traffic should always be routed to the sink node's CH, at which point the CH broadcasts data to the sink node and other cluster members. A source node with traffic to send always transmits to its CH. More specifically, communication paths are established between CHs and not individual sensor nodes.

The path from source node to the sink node's CH contains other CHs. Of those CHs, a subset broadcasts to their cluster members as well as the next hop CH. The election of broadcast CHs was discussed in Section A3 of this chapter.

To establish routing paths, we use Dijkstra's routing algorithm. Dijkstra's algorithm is a well-known, simple, least-cost algorithm that finds the lowest cost path from a source to a destination. Dijkstra's algorithm finds the shortest paths from a given source node to all other nodes by developing paths in order of increasing path length. Dijkstra's algorithm uses link costs to determine viable paths. Link costs are determined based on the network application. The first step of Dijkstra's algorithm is the initialization where a source node has the initial path costs to neighboring nodes. The second step is finding the next forwarding node beyond the neighboring nodes and recording the cost to get to it. The third step is updating the least cost path to each node, based on the information gained in step two [11]. For further details on Dijkstra, we refer the reader to [11].

We implement Dijkstra's algorithm at each CH to determine the least cost path to the sink node's CH. We used Euclidian distance as the cost between two CHs in Dijkstra's algorithm. The resulting path is the most energy efficient route through the WSN without factoring in the additional cost of the broadcast CHs.

As stated in Chapter I we aim to create an algorithm that is energy efficient. Implementing Dijkstra's algorithm carries a high initial energy cost due to the communications overhead necessary to establish the routes; however, our repeated use of the resulting least cost path yields energy savings that justify the upfront cost. The savings are obtained because all of the network traffic takes the least cost path when it is routed.

## C. SOURCE NODE TO SINK NODE PATH SUMMARY

When the network is deployed and initialized, the CHs are elected, the clusters are formed, broadcast CHs are determined, and the CHs implement Dijkstra's algorithm to find the least cost path from source to sink as described in the earlier sections. This process is illustrated in Figures 13, 14, 15 and 16.

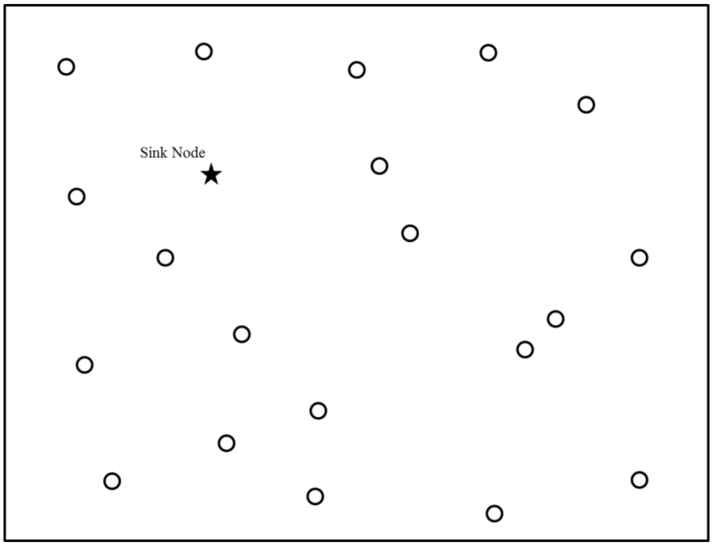


Figure 13. The WSN is deployed. All of the sensor nodes are placed randomly except the sink node, which is placed at (25 m, 75 m).

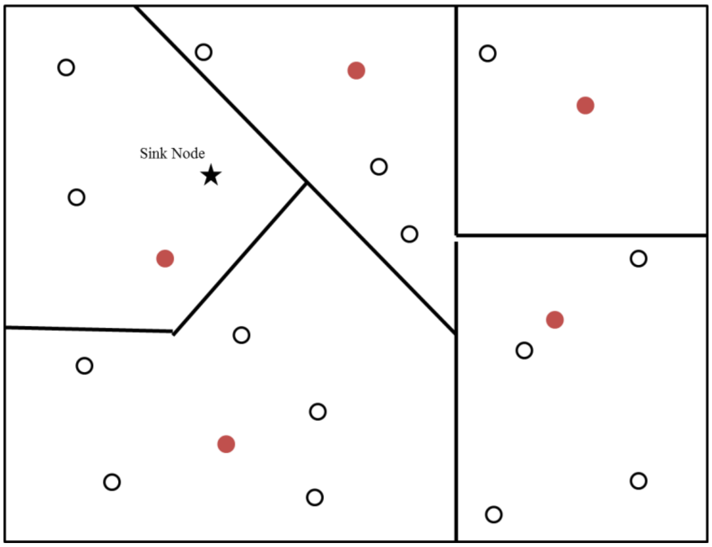


Figure 14. The WSN forms clusters with the election of CHs.

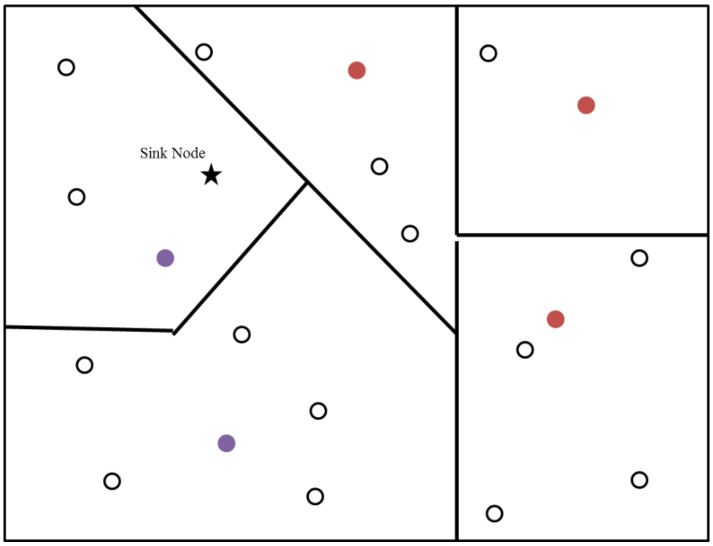


Figure 15. A subset of the CHs become broadcast CHs.

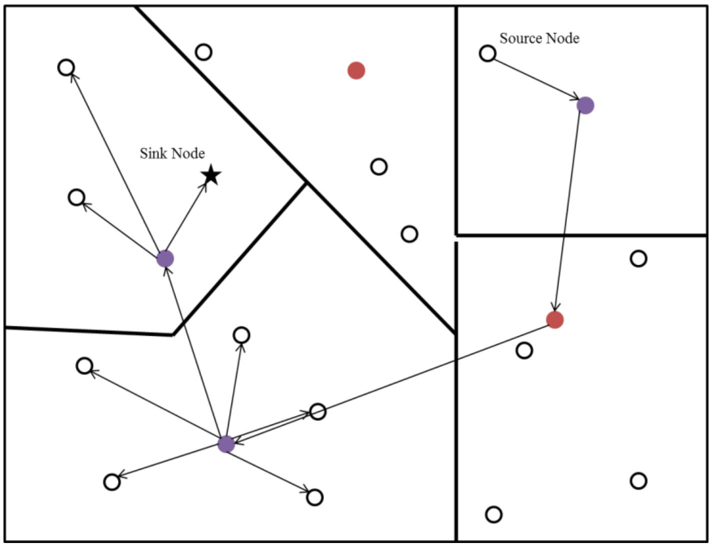


Figure 16. All cluster heads utilize Dijkstra's algorithm to determine the least cost route to the sink node's CH. Traffic is routing using the results of Dijkstra's algorithm. Broadcast CHs broadcast the data to all their cluster members.

When a source node senses information from its surrounding area, it processes the information and transmits the information to its CH. The CH is responsible for routing the message to the sink node's CH.

While transiting the route from source node's CH to sink node's CH, a subset of the intermediate CHs (the broadcast CHs) elect to broadcast the message.

When the sink node's CH receives a message destined for the sink node, it broadcasts the message to all of the sensor nodes which are cluster members.

## D. SINK NODE ANONYMITY

The goal of developing this algorithm is to ensure that at least *n* other nodes in the WSN have similar traffic statistics as the sink node. Let *N* be the set of all nodes in the WSN and let *i* denote the total number of nodes. In this thesis *i*=100 nodes: then,

$$N = \{n_1, n_2, .....n_i\}. \quad (1)$$

*CH* is the set of nodes which serve as CHs. The total number of CHs is denoted as *j*:

$$CH = \{\text{ch}_1, \text{ch}_2, .....\text{ch}_j\}. \quad (2)$$

*CM* is the set of nodes which serve as cluster members. The total number of cluster members is denoted as *k*:

$$CM = \{\mathrm{cm}_1, \mathrm{cm}_2, .....\mathrm{cm}_k\}\ . \tag{3}$$

At the end of the final iteration of CH election described in Section A1c of this chapter, all nodes in the WSN are either CHs or cluster members:

$$\mathrm{N} \equiv CH \cup CM \text{ and } i = j + k. \tag{4}$$

Each $n_i$ in *N* becomes an element of *CH* or *CM*:

$$n_i = ch_b \in CH \text{ or } n_i = cm_b \in CM\ . \tag{5}$$

The set *CH* is then ordered by the residual energy within each node:

$$CH_{energy} = \{\mathrm{ch}_5, \mathrm{ch}_8, .....\mathrm{ch}_j\}\ . \tag{6}$$

*BBCH* is the set of nodes which serve as broadcast CHs and is a subset of *CH*. The total number of broadcast CHs is denoted as *m*:

$$BCCH = \{\mathrm{bc}_1, \mathrm{bc}_2, .....\mathrm{bc}_m\} \text{ and } BBCH \subseteq CH\ . \tag{7}$$

Each broadcast CH is selected in order of maximum energy remaining: $bc_1 = ch_5$, $bc_2 = ch_8$ and so on. A broadcast CH broadcasts any data it receives to all of its cluster members in addition to the next hop CH. The total number of nodes broadcast to is denoted as β:

$$\beta = \sum_{i=1}^{m} \mathrm{members}(bc_i)\ . \tag{8}$$

The anonymity factor of the sink node is denoted as *AF* and is defined to be:

$$AF = 1/\beta \tag{9}$$

The number of cluster members that belong to each broadcast CH change each time the CHs are rotated. To evaluate the anonymity factor, we take the average value of the cluster members broadcast to across the simulation:

$$AF_{topo\mathrm{logy}} = 1/\,average(\beta)\ . \tag{10}$$

We use the preceding equations to evaluate the results of the simulations in Chapter VI.

## E. CHAPTER SUMMARY

In this chapter the cluster routing algorithm was introduced. The first step is the network initialization where nodes form clusters comprised of CHs and cluster members. A subset of broadcast CHs are also elected. The CHs utilized Dijkstra’s algorithm to find an energy efficient route from source node’s CH to the sink node’s CH. The method of evaluating the sink node’s anonymity was also introduced.

The solution proposed in this routing algorithm addresses sink node anonymity while being mindful of the energy consumption costs associated with any additional overhead incurred.

THIS PAGE INTENTIONALLY LEFT BLANK

# VI. SIMULATIONS AND RESULTS

In order to draw conclusions on the efficacy of the proposed anonymity algorithm, random network traffic must be simulated. A master file called simulations was developed to implement the network model, run simulated traffic over the modeled WSN and collect anonymity and energy metrics as results.

A trial is defined as one set of traffic messages that are routed across a topology. The message traffic was generated in four different volumes: 5,000, 10,000, 15,000 and 20,000 messages. In this thesis we generated four different physical topologies and conducted five trials using each traffic volume on each topology. A different set of traffic was randomly generated in each trial.

## A. SIMULATIONS

We created a number of files in MATLAB to develop the model on which to run these simulations and to generate the simulated traffic. They are categorized as follows.

### 1. WSN Topology Model

#### *a. Create_RandomSensorNetwork*

The *Create_RandomSensorNetwork* file randomly places all sensors inside a 100 meter by 100 meter square. The files utilizes the MATLAB function *rand*, which uses pseudo random variables drawn from the standard uniform distribution on the open interval (0,1). These values represent the $x$ and $y$ coordinates of each sensor node in meters in the WSN. The sink node is deliberately placed at the coordinate ($x$,$y$)= (25 m, 75 m). The location of the sink node within the WSN is not relevant, only that the node is deliberately placed. This aligns with military applications in which a WSN is tactically deployed, and the sink node is co-located with the personnel responsible for deploying the WSN. In this file the number of nodes, size of the geographic area, and location of the sink node can all be changed if desired.

#### *b. EnergyValues*

The *EnergyValues* file initializes the energy level for all sensors within the WSN. It also specifies transmit and receive communication costs and the processing or computational costs. The communication costs relative to sensing and processing costs were discussed in Chapter III. For simplification, sensing costs are declared minimal and are not included in the simulations. When the network is first deployed, all nodes have the same initial energy levels. The energy of each node is depleted by the network initialization and then the routing of network traffic.

The fixed transmission range from Chapter IV, Section A1 and fixed packet size simplify the communication costs. The transmit and receive communication costs are both fixed at $5.0\times10^{-7}$ W. The processing cost is $5.0\times10^{-8}$ W. The initial energy value of the nodes, transmit, receive and processing costs can all be changed in this file.

### 2. Clustering Algorithm

#### *a. ElectCH*

The *ElectCH* file is responsible for implementing the clustering protocol outlined in Chapter V. *ElectCH* carries out three iterations of electing CHs, but this can be modified to include additional iterations. The energy costs of electing CHs are also included within this file. The maximum transmission range and probability of a node becoming a CH can be changed in this file. *ElectCH* is also used when the CHs must be rotated in the network topology.

#### *b. CHadj*

The *CHadj* file creates an adjacency matrix for all of the CHs within the WSN. The matrix is reduced to contain only the CHs which are adjacent to one another within the maximum transmission range. The energy costs associated with this processing are also calculated and applied. The *CHadj* file returns a matrix called *adjCH*.

#### *c. CH_isotest*

The *CH_isotest* file evaluates an adjacency matrix to determine if any of the CH are isolated. That is, it determines if any CHs are not within the transmission range of any other CHs. This function is performed by the MATLAB *graphconncomp* function. If a CH is isolated, the *ElectCH* file is then applied and new CHs are elected. If new CHs are elected, the *CHadj* file is applied to the new CHs, and the *CH_isotest* is implemented again. This loop continues while there are isolated CHs. If there are no isolated CHs, the program continues. Again, the energy costs associated with this are accounted for.

### 3. Dijkstra Routing Algorithm

#### *a. Dij*

The *Dij* file implements Dijkstra's algorithm given an adjacency matrix, source node and destination node. Given these three inputs, the shortest path from source to destination is returned. The *Dij* code was taken from an existing research paper [26].

#### *b. CH_Route*

The *CH_Route* file utilizes the *Dij* file to ascertain and store the route from CH to sink node's CH  for each CH. The *Dij* program is executed at each CH. The route to the sink node's CH is then stored as *ClusterHead(i).Rte*. This route represents the path that network traffic takes across the WSN.

#### *c. Choose_BroadcastCH*

The *Choose_BroadcastCH* file determines how many cluster members each CH has and the amount of remaining energy in each CH. The CH with the largest amount of energy becomes a broadcast CH, and the number of its cluster members is saved. Additional CHs become broadcast CHs based on their energy levels, with those with the most energy being added first. This continues until at least 20 cluster members are being broadcast to. That is, the sum of the number of the cluster members of all of the elected broadcast CHs exceeds 20. If the sink node's CH is not chosen to be a broadcast CH based on its energy value, it is added to the set of broadcast CHs, and its members are added to the total number of nodes broadcast to.

### 4. Traffic Generation and Simulation

#### *a. SourceSim*

The *SourceSim* file utilizes the MATLAB *randi* function to generate a source node matrix. Each entry in the matrix represents a source node that routes traffic to the sink node. The network model specifies that the WSN utilizes event driven reporting so each value of the *SourceSim* file is executed sequentially. The total number of messages is changed throughout the simulations to 5,000, 10,000, 15,000 and 20,000 messages and can be changed in this file to other values.

#### *b. Sim_Loop_2*

The *Sim_Loop_2* file routes the traffic from *SourceSim* across the WSN. The traffic is routed across the network, and the energy value of all of the intermediate nodes is decremented. When the conditions to rotate the CHs are met, as outlined in Chapter V, Section A2, *Sim_Loop_2* rotates the CHs and continues routing traffic until all of the traffic has been routed through the network.

### 5. Collecting Results

#### *a. Energy_Metrics*

The *Energy_Metrics* file returns the maximum, minimum, and average energy consumed by the nodes in the WSN. The index of the maximum and minimum energy node is also returned. The index of the maximum and minimum energy node is compared to the source matrix traffic, returning the number of times the maximum and minimum energy nodes route traffic through the WSN over the trial.

#### *b. Anony_Metrics*

The *Anony_Metrics* file records the total number of nodes broadcast to and the number of members of the sink node's CH for each rotation of the CHs and returns the average number of nodes broadcast to and the average number of nodes in the sink node's CH. The average number of nodes broadcast to is used in Eq. (8) and Eq. (9) to calculate the anonymity.

### 6. House Keeping Files

#### *a. SaveEnergyValues*

The *SaveEnergyValues* file updates the energy value of each sensor on the master list. This is a function of the code and MATLAB but is not a step that needs to be executed in an actual implementation as the nodes always know their energy values. *SaveEnergyValues* moves the energy values from *ClusterMember(i).E* to *N(i).E*. *SaveEnergyValues* is used immediately prior to re-electing CHs each time.

#### *b. Plot_Results*

The *Plot_Results* file returns four figures which are plots of the WSN and the iterations of electing CHs. These plots are not a functional part of the routing of network traffic but provide a visual representation of the network for the purposes of this thesis. *Plot_Results* uses different symbols to represent the CHs which are elected in every iteration.

## B. ANALYSIS OF RESULTS

### 1. Topology 1

The physical topology of Topology 1 was generated using the *Create_RandomSensorNetwork* discussed in Section A1 of this chapter. The physical location of the nodes remains the same throughout Topology 1. Across the five trials at each simulated traffic volume, the only thing that changes is the role each nodes plays in the WSN.

#### *a. Energy Results*

The simulations described in Section A of this chapter are conducted over Topology 1, and the results are summarized in Table 3. This information is used to generate Figures 17, 18 and 19.

Table 3. The average, maximum, and minimum energy consumed by nodes in Topology 1 over five trials at each traffic volume. The roles played by the maximum energy node contributed to an understanding of what drives the energy consumption of the MaxECN.

| | 5000 Messages | | | | |
|---|---|---|---|---|---|
| Trial | 1 | 2 | 3 | 4 | 5 |
| Energy | | | | | |
| Average Energy Expended by a Node | 4.0436x10$^{-4}$ | 3.5282x10$^{-4}$ | 4.0187x10$^{-4}$ | 3.7404x10$^{-4}$ | 3.4066x10$^{-4}$ |
| Min Energy Expended By a Node | 1.7900x10$^{-5}$ | 1.6750x10$^{-5}$ | 2.4300x10$^{-5}$ | 1.4200x10$^{-5}$ | 1.6550x10$^{-5}$ |
| Max Energy Expended By a Node | 7.3000x10$^{-3}$ | 5.1000x10$^{-3}$ | 5.1000x10$^{-3}$ | 3.8000x10$^{-3}$ | 5.7000x10$^{-3}$ |
| Distance of Max Energy Node from Sink Node | 10.4169 | 10.4169 | 14.9503 | 14.9503 | 7.0315 |
| # of Messages Max Energy Node Sends | 57 | 41 | 52 | 57 | 53 |
| # of times Max Energy Node is a Cluster Member | 3 | 5 | 5 | 3 | 3 |
| # of times Max Energy Node is a Cluster Head | 3 | 1 | 1 | 2 | 2 |
| # of times Max Energy Node is a Broadcast Cluster Head | 3 | 1 | 1 | 1 | 2 |
| # of times Max Energy Node is Sink Node Cluster Head | 3 | 1 | 1 | 1 | 2 |
| | 10,000 Messages | | | | |
| Trial | 1 | 2 | 3 | 4 | 5 |
| Energy | | | | | |
| Average Energy Expended by a Node | 7.6265x10$^{-4}$ | 7.2387x10$^{-4}$ | 7.2830x10$^{-4}$ | 7.1345x10$^{-4}$ | 8.0101x10$^{-4}$ |
| Min Energy Expended By a Node | 3.4600x10$^{-5}$ | 3.5600x10$^{-5}$ | 4.9000x10$^{-5}$ | 3.2100x10$^{-5}$ | 3.6850x10$^{-5}$ |
| Max Energy Expended By a Node | 8.5000x10$^{-3}$ | 1.1500x10$^{-2}$ | 7.0000x10$^{-3}$ | 1.2800x10$^{-2}$ | 7.8000x10$^{-3}$ |
| Distance of Max Energy Node from Sink Node | 17.2641 | 10.4169 | 10.4169 | 10.4169 | 10.4169 |
| # of Messages Max Energy Node Sends | 119 | 96 | 83 | 111 | 83 |
| # of times Max Energy Node is a Cluster Member | 7 | 6 | 6 | 7 | 8 |
| # of times Max Energy Node is a Cluster Head | 4 | 5 | 5 | 4 | 3 |
| # of times Max Energy Node is a Broadcast Cluster Head | 2 | 5 | 4 | 4 | 3 |
| # of times Max Energy Node is Sink Node Cluster Head | 2 | 5 | 4 | 4 | 2 |
| | 15,000 Messages | | | | |
| Trial | 1 | 2 | 3 | 4 | 5 |
| Energy | | | | | |
| Average Energy Expended by a Node | 1.2000x10$^{-3}$ | 1.1000x10$^{-3}$ | 1.1000x10$^{-3}$ | 1.1000x10$^{-3}$ | 1.2000x10$^{-3}$ |
| Min Energy Expended By a Node | 4.4450x10$^{-5}$ | 4.8350x10$^{-5}$ | 4.6300x10$^{-5}$ | 4.9850x10$^{-5}$ | 5.5300x10$^{-5}$ |
| Max Energy Expended By a Node | 1.0400x10$^{-2}$ | 9.9000x10$^{-3}$ | 1.1500x10$^{-2}$ | 1.4200x10$^{-2}$ | 1.4400x10$^{-2}$ |
| Distance of Max Energy Node from Sink Node | 12.9479 | 7.0315 | 7.0315 | 7.0315 | 12.9479 |
| # of Messages Max Energy Node Sends | 142 | 174 | 148 | 134 | 171 |
| # of times Max Energy Node is a Cluster Member | 11 | 12 | 11 | 10 | 12 |
| # of times Max Energy Node is a Cluster Head | 5 | 4 | 5 | 6 | 4 |
| # of times Max Energy Node is a Broadcast Cluster Head | 4 | 4 | 5 | 6 | 3 |
| # of times Max Energy Node is Sink Node Cluster Head | 4 | 4 | 5 | 6 | 3 |
| | 20,000 Messages | | | | |
| Trial | 1 | 2 | 3 | 4 | 5 |
| Energy | | | | | |
| Average Energy Expended by a Node | 1.5000x10$^{-3}$ | 1.5000x10$^{-3}$ | 1.6000x10$^{-3}$ | 1.5000x10$^{-3}$ | 1.4000x10$^{-3}$ |
| Min Energy Expended By a Node | 6.3300x10$^{-5}$ | 6.4350x10$^{-5}$ | 6.3050x10$^{-5}$ | 6.8200x10$^{-5}$ | 5.6700x10$^{-5}$ |
| Max Energy Expended By a Node | 1.2000x10$^{-2}$ | 1.3700x10$^{-2}$ | 1.5300x10$^{-2}$ | 1.4000x10$^{-2}$ | 1.3900x10$^{-2}$ |
| Distance of Max Energy Node from Sink Node | 16.4091 | 7.0315 | 10.4169 | 12.9479 | 7.0315 |
| # of Messages Max Energy Node Sends | 244 | 219 | 208 | 195 | 196 |
| # of times Max Energy Node is a Cluster Member | 14 | 14 | 14 | 13 | 14 |
| # of times Max Energy Node is a Cluster Head | 7 | 7 | 7 | 9 | 7 |
| # of times Max Energy Node is a Broadcast Cluster Head | 2 | 7 | 6 | 6 | 7 |
| # of times Max Energy Node is Sink Node Cluster Head | 2 | 7 | 6 | 6 | 7 |

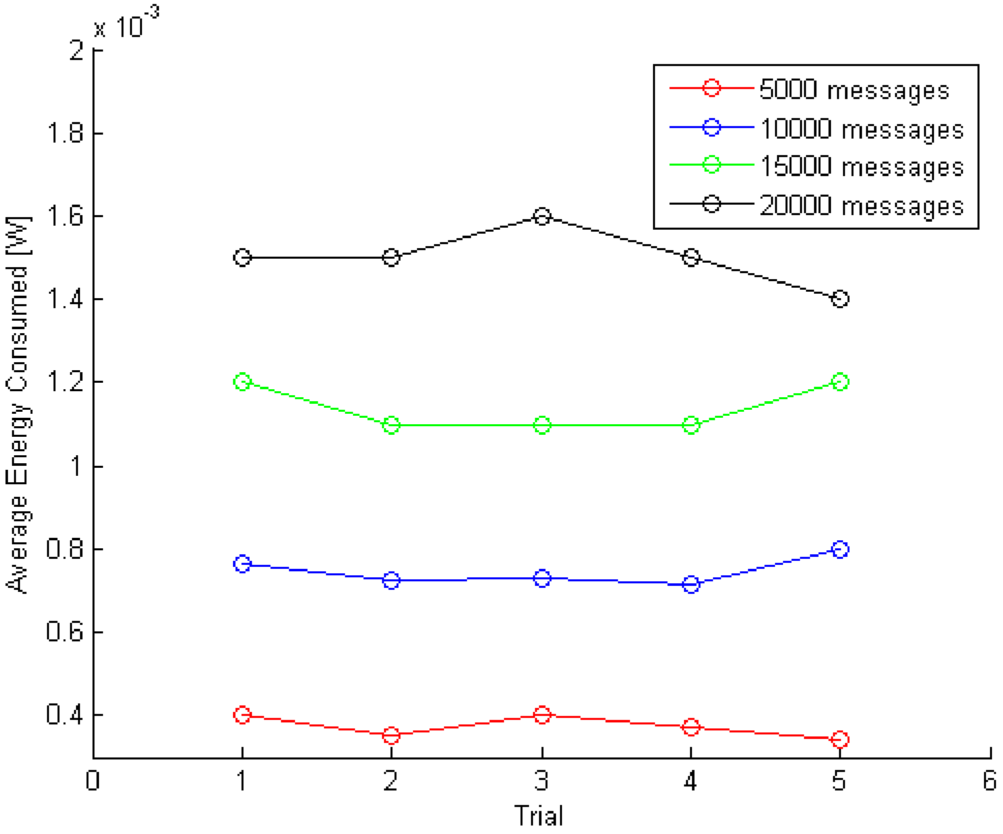


Figure 17. The average energy consumed by the nodes in the WSN for 5,000, 10,000, 15,000, and 20,000 messages. The average energy consumed increases as traffic volume increases in all five trials in Topology 1.

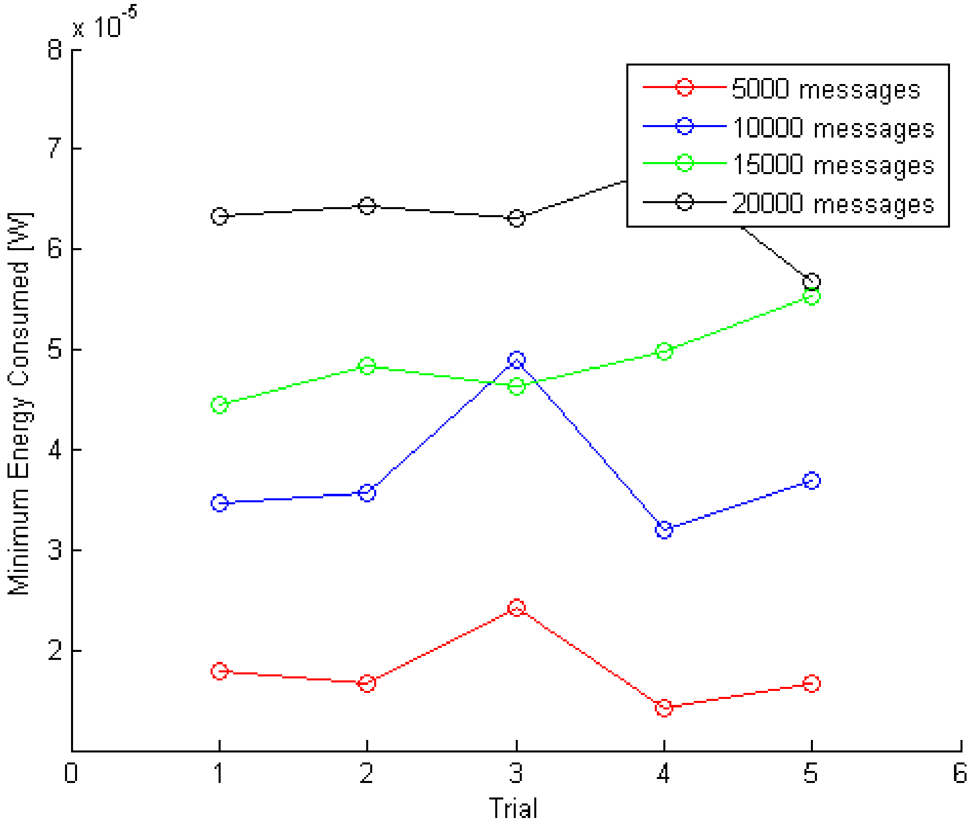


Figure 18. The minimum energy consumed by nodes in the WSN for 5,000, 10,000, 15,000, and 20,000 messages. The minimum energy consumed by a node increases as traffic volume increases in Topology 1. At trial 3, 10,000 messages, MinEC deviates from the average MinEC at that traffic volume.

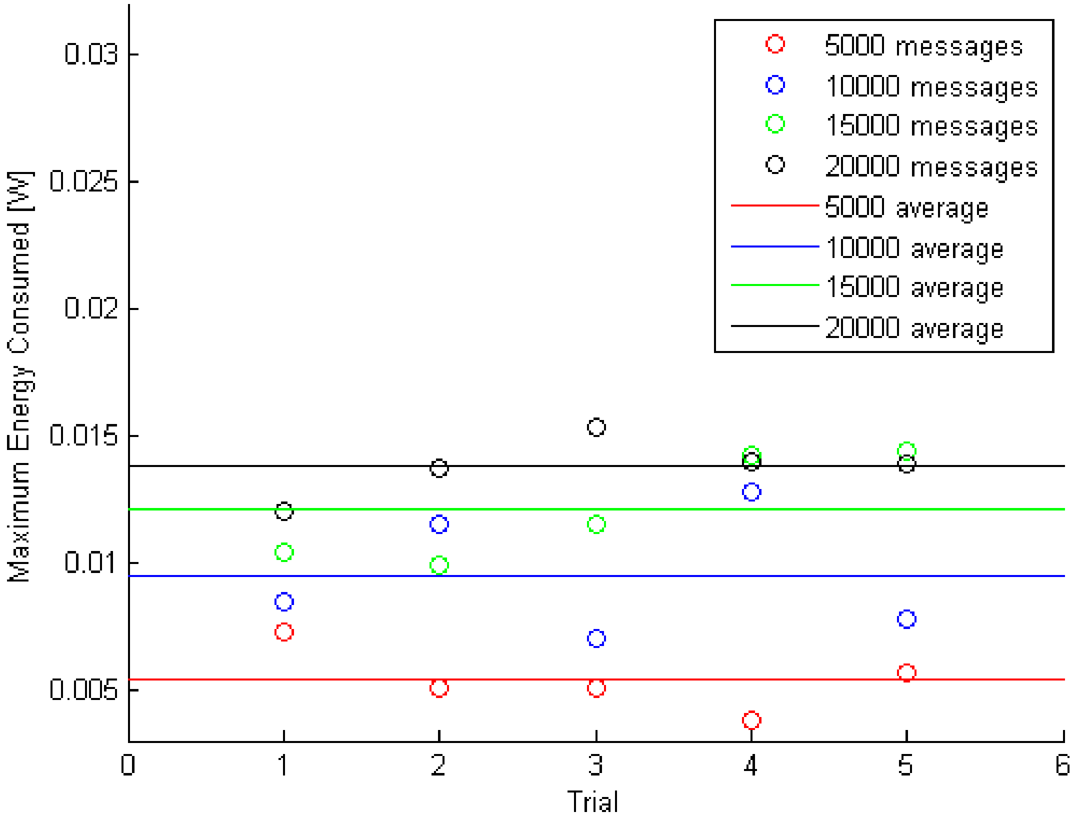


Figure 19. The maximum energy consumed by nodes in the WSN for 5,000, 10,000, 15,000, and 20,000 messages. The maximum energy consumed increases as traffic volume increases in all five trials but the values are subject to overlapping across message volumes in Topology 1.

We see in Figure 17 that the average energy consumed (AvgEC) by a node in the WSN is consistent across the five trials and increases with the increase in traffic volume across the WSN.

The minimum energy consumed by a node (MinEC) is shown in Figure 18. The results for the minimum energy follow a similar pattern of increasing when the traffic volume increases. There is one anomaly at trial 3, 10,000 messages, where the minimum energy consumed at 10,000 messages exceeds the minimum energy consumed at 15,000 messages. Examining Figure 19, we see that in trial 3, 10,000 messages, the maximum energy consumed node is the smallest value among the five trials. We conclude that in trial 3, 10,000 messages, the minimum and maximum values for energy consumed were simply closer to the average than in other trials as the average in Figure 17 is consistent with the other values.

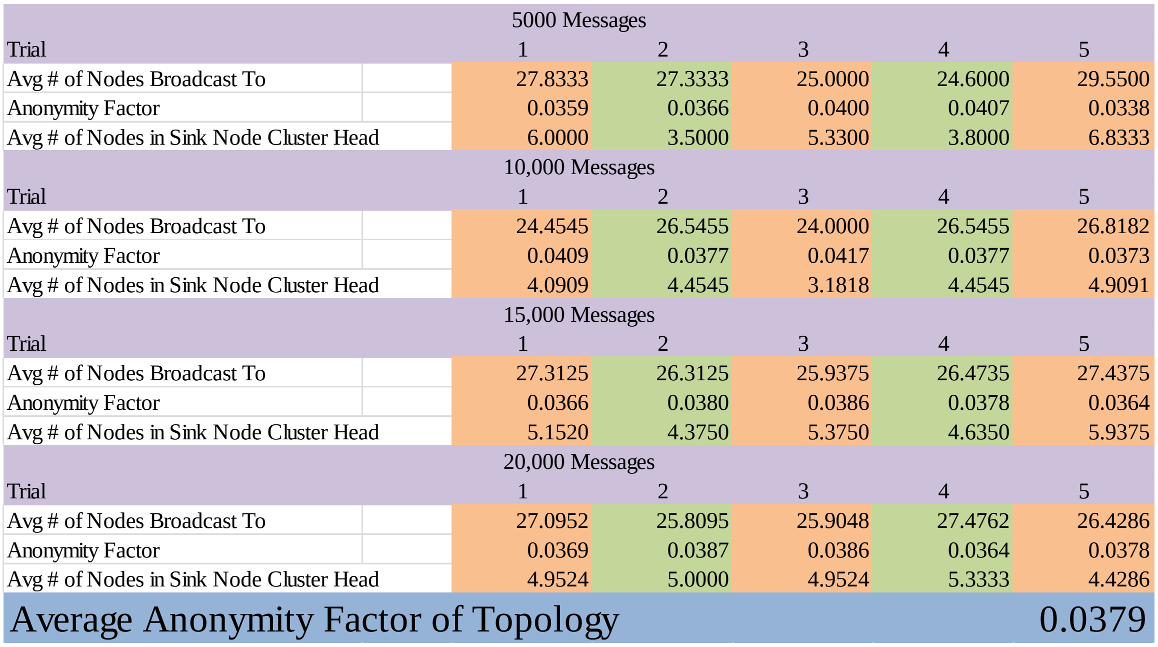
We see in Figure 19 that while the general trend of increased consumption with increased traffic volume holds, the values of each trial fluctuate significantly. The maximum energy consumed by a node (MaxEC) in the WSN has far less predictable results than the average and minimum energy values.

The numerical values for the average, minimum, and maximum energy consumed by nodes in the WSN are shown in Table 3. From this table it can be seen that in trial 1, 20,000 messages, the MaxEC is equal to the average MaxEC of 15,000 messages across all five trials. We see in Table 3 that in trial 1 the number of times the maximum energy consumed node (MaxECN) serves as a broadcast CH or the sink node's CH is two, compared to six or seven times for the other four trials. Thus, the MaxEC is well below the average MaxEC for 20,000 messages.

For trial 2, 15,000 messages, the maximum energy consumed is less than trial 2, 10,000 messages and is near the average MaxEC for 10,000 messages. The number of messages the maximum energy consumed node (MaxECN) sends through the WSN is typical when compared to other values at the same traffic volume. The number of times the MaxECN serves as a CH, broadcast CH and the sink node's CH is the highest of the five trials when 10,000 messages are sent, driving the MaxEC higher. The MaxECN in trial 2, 15,000 messages, sends the second highest number of messages through the WSN and performs the role of cluster member, CH, broadcast CH, and sink node CH a similar number of times as its peers but has a lower MaxEC. We can attribute this to one of two factors. The first is that the network initialization of the topology was more efficient in this trial, keeping the MaxEC lower. The source of this efficiency is *CH_isotest*, which was discussed in Chapter VI, Section A2. If the topology fails the isolation test and must re-elect cluster heads, this imposes an additional energy costs across the WSN. If, as the CHs rotate, the topology never fails the isolation test, then the energy consumption is lower across the WSN. The other option is that the number of nodes broadcast to when the MaxECN served as a broadcast CH and as the sink node's CH was less than in other trials; therefore, the MaxECN was lower.

In trial 4, 15,000 messages, the maximum energy consumed is greater than trial 4 and 20,000 messages and above the 20,000 message average. The number of messages the MaxECN sends through the WSN is the lowest of all five trials. However, the MaxECN serves as a CH, broadcast CH and sink node's CH more times than in any other trial. These roles require more energy than being a cluster member, so the MaxEC for this trial is driven up.

Table 4. The total number of nodes broadcast to and the anonymity factor for each trial and traffic volume of Topology 1.



| | 5000 Messages | | | | |
|---|---|---|---|---|---|
| Trial | 1 | 2 | 3 | 4 | 5 |
| Avg # of Nodes Broadcast To | 27.8333 | 27.3333 | 25.0000 | 24.6000 | 29.5500 |
| Anonymity Factor | 0.0359 | 0.0366 | 0.0400 | 0.0407 | 0.0338 |
| Avg # of Nodes in Sink Node Cluster Head | 6.0000 | 3.5000 | 5.3300 | 3.8000 | 6.8333 |
| | 10,000 Messages | | | | |
| Trial | 1 | 2 | 3 | 4 | 5 |
| Avg # of Nodes Broadcast To | 24.4545 | 26.5455 | 24.0000 | 26.5455 | 26.8182 |
| Anonymity Factor | 0.0409 | 0.0377 | 0.0417 | 0.0377 | 0.0373 |
| Avg # of Nodes in Sink Node Cluster Head | 4.0909 | 4.4545 | 3.1818 | 4.4545 | 4.9091 |
| | 15,000 Messages | | | | |
| Trial | 1 | 2 | 3 | 4 | 5 |
| Avg # of Nodes Broadcast To | 27.3125 | 26.3125 | 25.9375 | 26.4735 | 27.4375 |
| Anonymity Factor | 0.0366 | 0.0380 | 0.0386 | 0.0378 | 0.0364 |
| Avg # of Nodes in Sink Node Cluster Head | 5.1520 | 4.3750 | 5.3750 | 4.6350 | 5.9375 |
| | 20,000 Messages | | | | |
| Trial | 1 | 2 | 3 | 4 | 5 |
| Avg # of Nodes Broadcast To | 27.0952 | 25.8095 | 25.9048 | 27.4762 | 26.4286 |
| Anonymity Factor | 0.0369 | 0.0387 | 0.0386 | 0.0364 | 0.0378 |
| Avg # of Nodes in Sink Node Cluster Head | 4.9524 | 5.0000 | 4.9524 | 5.3333 | 4.4286 |
| Average Anonymity Factor of Topology | | | | | 0.0379 |

Trial 5, 15,000 messages, also exceeds the 20,000 message MaxEC average. The roles of the MaxECN of trial 5, 15,000 messages, are very similar to trial 2, 15,000 messages, and yet the MaxEC for trial 5 is highest of the five trials, while the MaxEC for trial 2 is the lowest. Looking into Table 4, we see that the average number of nodes in the sink node's CH is at a maximum for trial 5 at 5.9375 and at a minimum for trial 2 at 4.3750. We conclude that in this case the MaxEC is driven higher by the costs of the broadcasting to a larger number sink node cluster members.

### *b. Sink Node Anonymity*

The anonymity factor is calculated based on the total number of nodes broadcast to, as outlined in Chapter V, Section D. The results returned from *Anony_Metrics* are listed in Table 4 in the preceding subsection and used to calculate the anonymity factor for each trial and traffic volume. The average number of nodes broadcast to ranges from 24.0000 to 29.5550 and exceeds the desired threshold of 20 nodes. The results vary based on traffic volume and do not demonstrate any trends of convergence to a number of nodes broadcast to or divergence from a number of nodes broadcast to as traffic volume increases, is shown in Figure 20. Just as there are no trends in the average number of nodes broadcast to, there are no trends on the anonymity factor over the different traffic volumes. This is shown in Figure 21. By taking the average of all of the anonymity factors calculated in Table 4, we calculate to average anonymity factor of the topology to be 0.0379.

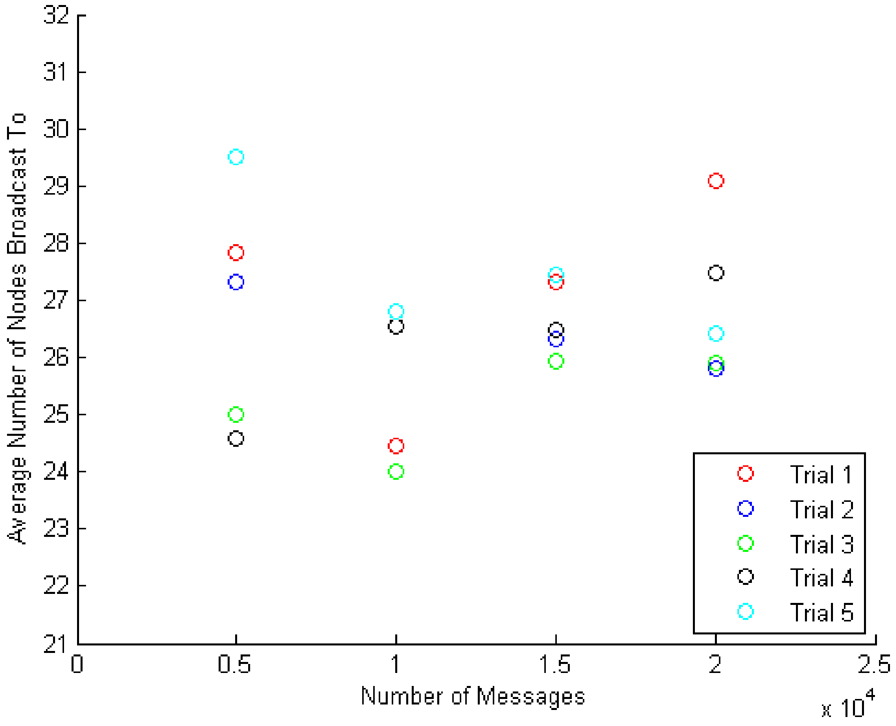


Figure 20. The average number of nodes broadcast to for different traffic volumes over five trials. The average number of nodes broadcast to is between 24.0000 and 29.5500 for Topology 1.

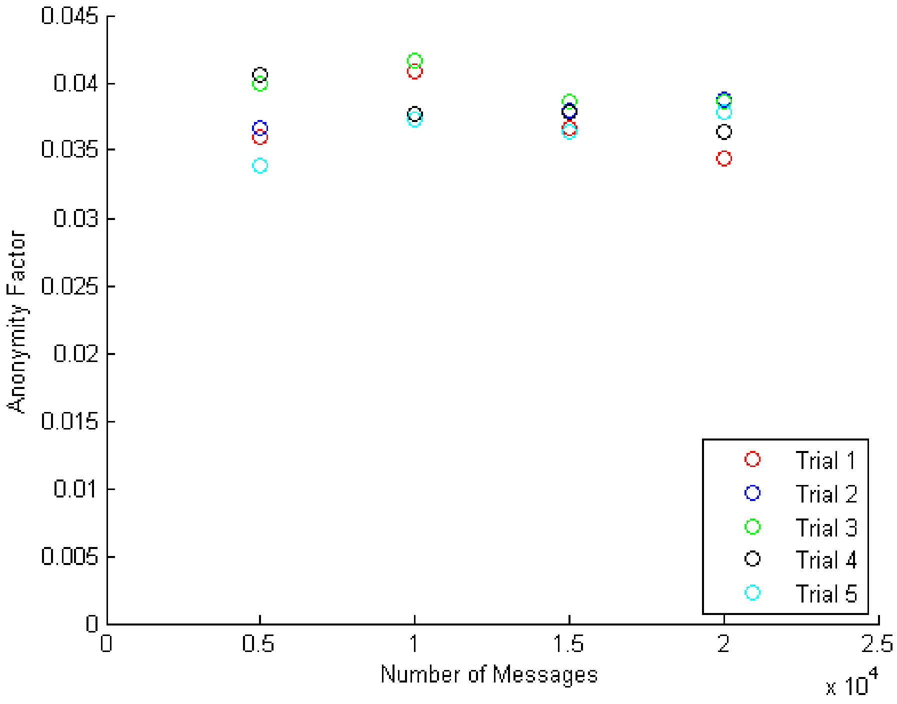


Figure 21. The anonymity factor of each trial at each traffic volume for Topology 1.

### 2. Topology 2

Just as in Topology 1, the physical topology of Topology 2 was generated using the *Create_RandomSensorNetwork* discussed in Section A1 of this chapter. The physical location of the nodes remains the same throughout Topology 2. Across the five trials at each simulated traffic length, the only thing that changes is the role each nodes plays in the WSN.

#### *a. Energy Results*

The simulations described in Section A of this chapter are conducted over Topology 2, and the results are summarized in Table 5. This information is used to generate Figures 22, 23, and 24.

Table 5. The average, maximum, and minimum energy consumed by nodes in Topology 2 over five trials at each traffic volume. The roles played by the maximum energy node contributed to an understanding of what drives the energy consumption of the MaxECN.

| | | | 5000 Messages | | |
|---|---|---|---|---|---|
| Trial | 1 | 2 | 3 | 4 | 5 |
| Energy | | | | | |
| Average Energy Expended by a Node | 4.0533x10$^{-4}$ | 5.5851x10$^{-4}$ | 4.0151x10$^{-4}$ | 3.6524x10$^{-4}$ | 3.9052x10$^{-4}$ |
| Min Energy Expended By a Node | 1.3700x10$^{-5}$ | 2.1200x10$^{-5}$ | 1.6100x10$^{-5}$ | 1.6050x10$^{-5}$ | 1.9450x10$^{-5}$ |
| Max Energy Expended By a Node | 9.8000x10$^{-3}$ | 1.0200x10$^{-2}$ | 9.7000x10$^{-3}$ | 4.1000x10$^{-3}$ | 1.3400x10$^{-2}$ |
| Distance of Max Energy Node from Sink Node | 5.3585 | 5.4363 | 5.4363 | 4.6758 | 4.6758 |
| # of Messages Max Energy Node Sends | 48 | 48 | 58 | 51 | 37 |
| # of times Max Energy Node is a Cluster Member | 2 | 4 | 3 | 4 | 2 |
| # of times Max Energy Node is a Cluster Head | 4 | 3 | 3 | 1 | 4 |
| # of times Max Energy Node is a Broadcast Cluster Head | 3 | 2 | 2 | 1 | 4 |
| # of times Max Energy Node is Sink Node Cluster Head | 3 | 2 | 2 | 1 | 4 |
| | | | 10,000 Messages | | |
| Trial | 1 | 2 | 3 | 4 | 5 |
| Energy | | | | | |
| Average Energy Expended by a Node | 8.0955x10$^{-4}$ | 8.1107x10$^{-4}$ | 9.0089x10$^{-4}$ | 7.7364x10$^{-4}$ | 8.8602x10$^{-4}$ |
| Min Energy Expended By a Node | 3.1150x10$^{-5}$ | 3.1750x10$^{-5}$ | 2.7100x10$^{-5}$ | 3.1800x10$^{-5}$ | 2.8650x10$^{-5}$ |
| Max Energy Expended By a Node | 7.4000x10$^{-3}$ | 1.6400x10$^{-2}$ | 1.9800x10$^{-2}$ | 1.5000x10$^{-2}$ | 1.4200x10$^{-2}$ |
| Distance of Max Energy Node from Sink Node | 15.0279 | 2.2242 | 11.2799 | 4.6758 | 2.2242 |
| # of Messages Max Energy Node Sends | 100 | 91 | 97 | 107 | 92 |
| # of times Max Energy Node is a Cluster Member | 7 | 7 | 6 | 6 | 8 |
| # of times Max Energy Node is a Cluster Head | 4 | 4 | 6 | 5 | 3 |
| # of times Max Energy Node is a Broadcast Cluster Head | 2 | 4 | 4 | 5 | 3 |
| # of times Max Energy Node is Sink Node Cluster Head | 2 | 4 | 4 | 5 | 3 |
| | | | 15,000 Messages | | |
| Trial | 1 | 2 | 3 | 4 | 5 |
| Energy | | | | | |
| Average Energy Expended by a Node | 1.3000x10$^{-3}$ | 1.3000x10$^{-3}$ | 1.3000x10$^{-3}$ | 1.2000x10$^{-3}$ | 1.3000x10$^{-3}$ |
| Min Energy Expended By a Node | 4.5400x10$^{-5}$ | 4.4150x10$^{-5}$ | 4.2400x10$^{-5}$ | 3.8300x10$^{-5}$ | 4.6100x10$^{-5}$ |
| Max Energy Expended By a Node | 1.3900x10$^{-2}$ | 2.3200x10$^{-2}$ | 2.2000x10$^{-2}$ | 1.1600x10$^{-2}$ | 1.1300x10$^{-2}$ |
| Distance of Max Energy Node from Sink Node | 4.6758 | 5.3585 | 5.3585 | 5.3585 | 4.6758 |
| # of Messages Max Energy Node Sends | 137 | 139 | 142 | 143 | 154 |
| # of times Max Energy Node is a Cluster Member | 12 | 9 | 12 | 12 | 11 |
| # of times Max Energy Node is a Cluster Head | 5 | 8 | 5 | 5 | 5 |
| # of times Max Energy Node is a Broadcast Cluster Head | 4 | 5 | 5 | 4 | 3 |
| # of times Max Energy Node is Sink Node Cluster Head | 4 | 5 | 5 | 4 | 3 |
| | | | 20,000 Messages | | |
| Trial | 1 | 2 | 3 | 4 | 5 |
| Energy | | | | | |
| Average Energy Expended by a Node | 1.5000x10$^{-3}$ | 1.6000x10$^{-3}$ | 1.8000x10$^{-3}$ | 1.6000x10$^{-3}$ | 1.6000x10$^{-3}$ |
| Min Energy Expended By a Node | 6.1450x10$^{-5}$ | 6.0250x10$^{-5}$ | 5.8550x10$^{-5}$ | 5.9950x10$^{-5}$ | 4.6950x10$^{-5}$ |
| Max Energy Expended By a Node | 1.7300x10$^{-2}$ | 2.2700x10$^{-2}$ | 1.7500x10$^{-2}$ | 2.0200x10$^{-2}$ | 2.2400x10$^{-2}$ |
| Distance of Max Energy Node from Sink Node | 5.4363 | 5.3585 | 11.2799 | 4.6758 | 6.4270 |
| # of Messages Max Energy Node Sends | 210 | 208 | 185 | 194 | 231 |
| # of times Max Energy Node is a Cluster Member | 14 | 13 | 17 | 15 | 16 |
| # of times Max Energy Node is a Cluster Head | 7 | 9 | 6 | 7 | 5 |
| # of times Max Energy Node is a Broadcast Cluster Head | 6 | 6 | 4 | 5 | 5 |
| # of times Max Energy Node is Sink Node Cluster Head | 5 | 6 | 4 | 5 | 5 |

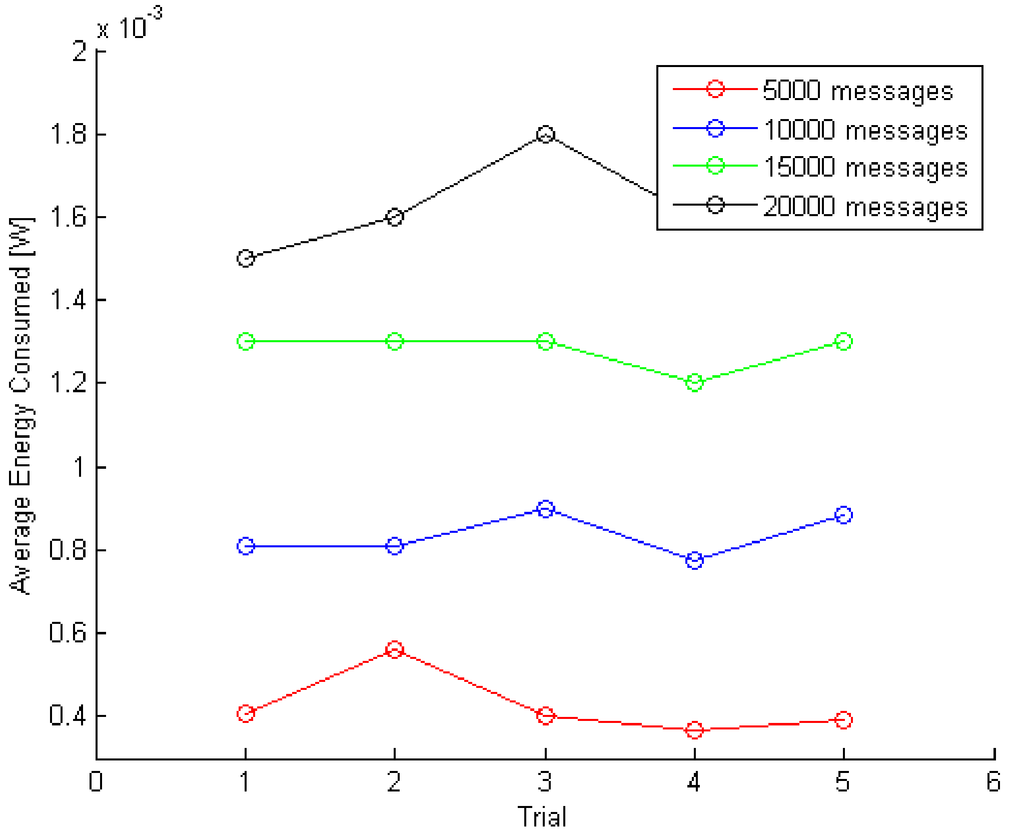


Figure 22. The average energy consumed by nodes in the WSN for 5,000, 10,000, 15,000, and 20,000 messages. The average energy consumed increases as traffic volume increases in all five trials of Topology 2.

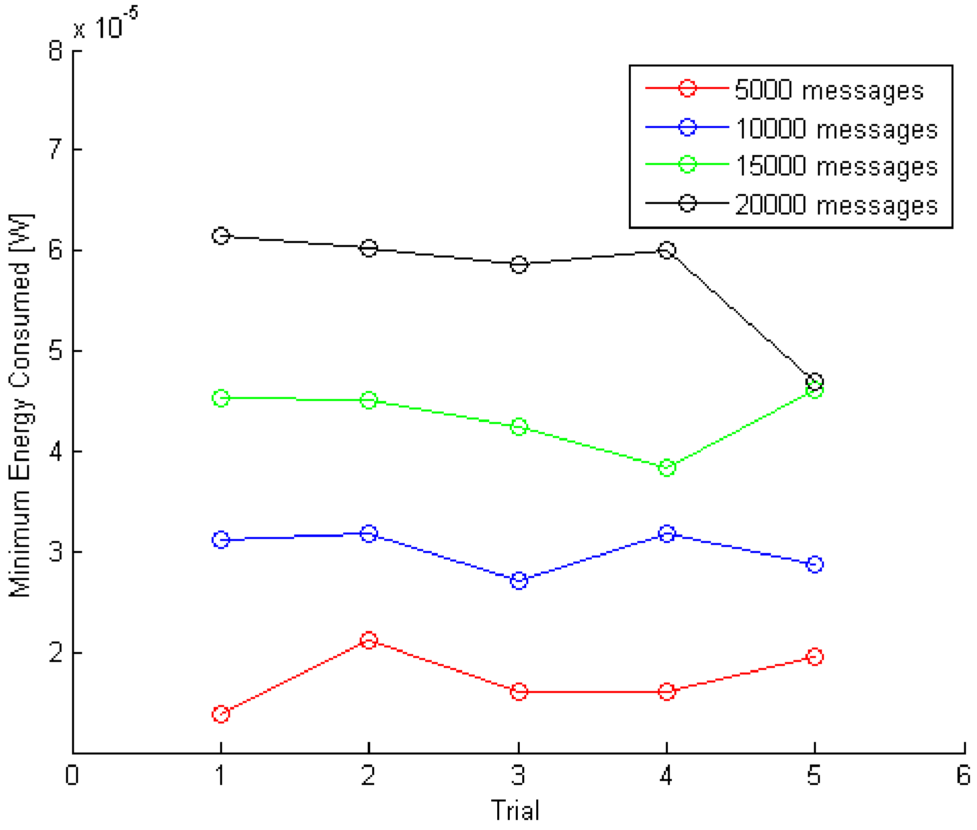


Figure 23. The minimum energy consumed by nodes in the WSN for 5,000, 10,000, 15,000, and 20,000 messages. The minimum energy consumed increases as traffic volume increases in all five trials in Topology 2.

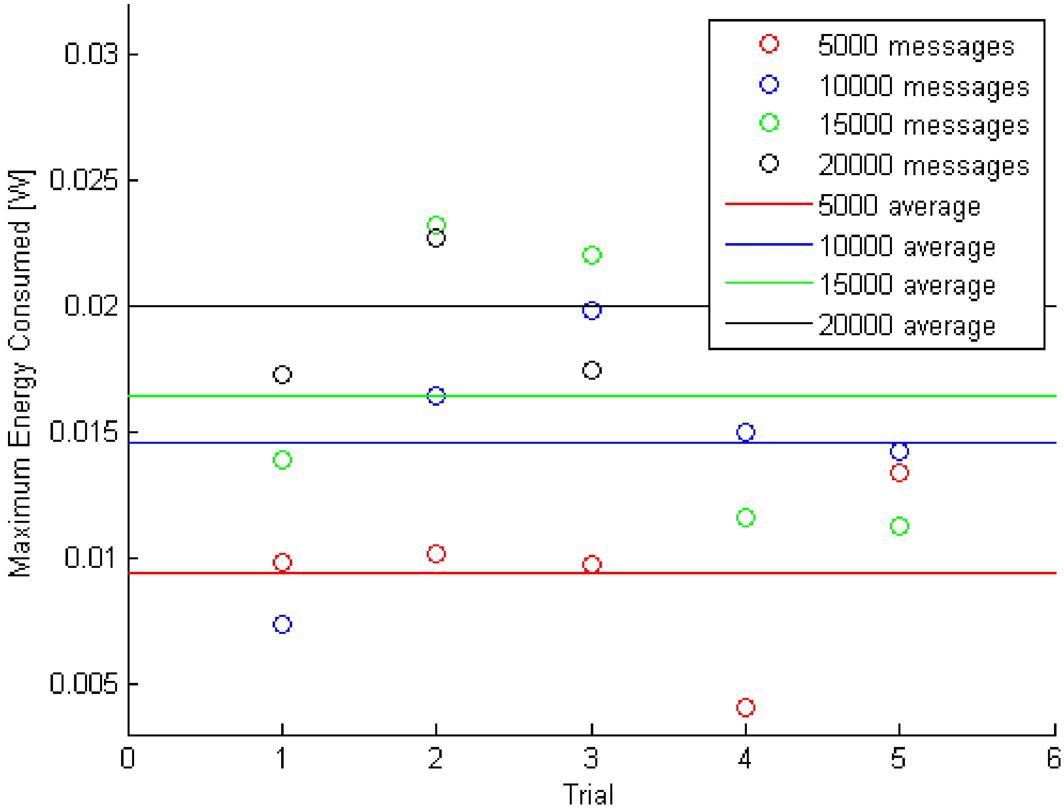


Figure 24. The maximum energy consumed by nodes in the WSN for 5,000, 10,000, 15,000, and 20,000 messages. The maximum energy consumed increases as traffic volume increases in all five trials. The values across the five trials fluctuate in Topology 2.

The AvgEC by a node in the WSN is consistent across the five trials and increases with the increase in traffic volume across the WSN, as shown in Figure 22. The MinEC also follows the same trend, increasing with the increase in traffic volume, though the values fluctuate more within the five trials at each traffic volume. This is shown in Figure 23.

Just as in Topology 1, the maximum energy consumed increases as traffic volume increases in all five trials, but the values are subject to overlapping across traffic volumes.

The MaxEC of trial 1, 10,000 messages is below the average MaxEc for 5,000 messages, as shown in Figure 24. The MaxECN serves as a CH a similar number of times compared to the other trials; however, it only spends half the time as a broadcast or sink node’s CH, causing the MaxEC to be much lower than the average for this traffic volume in Topology 2. In trial 1, 15,000 messages, the MaxEC is below the average MaxEC for 10,000 messages. The roles played by the MaxECN are similar to the rest of the trials and the average number of nodes broadcast to is in line with the other trials at this traffic volume. The average energy consumed by a node, illustrated in Figure 22, is on par with all other trials at this traffic volume, so we conclude that while below the average for 10,000 messages, nothing is abnormal in this trial. The MaxECN simply consumed less energy.

In trial 2, 15,000 messages, the MaxECN serves as a CHs eight times, which is three more times than any other trial. This contributes to a higher than average MaxEC for that trial and slightly exceeds the MaxEC of trial 2, 20,000 messages. The MaxEC of trial 2, 10,000 messages is equal to the average MaxEC for 15,000 messages. Examining the role of the MaxECN is this trial compared to other trials and the average number of nodes broadcast to, we see that there is not a specific factor for this trial that is driving the MaxEC to the average of the MaxEC for 15,000 messages. We simply conclude that, while the average energy consumed by this trial is similar to the other trials at this traffic volume, this trial was less energy efficient in terms of the maximum energy consumed by a node.

In trial 3, 10,000 messages, the MaxEC is equal to that of the average MaxEC of 20,000 messages. Examining the data in Table 6, we see that the roles performed by the MaxEC are similar to the other trials at this traffic volume. The number of nodes broadcast to is the lowest of the five trials for 10,000 messages, so it is not contributing to the larger MaxEC either.  We see in Figure 19 that the average energy consumed by a node in the WSN is also highest at this trial. Thus, we conclude that the WSN initialization drove the energy costs up for the trial. The MaxEC of trial 3, 15,000 messages exceeds the average MaxEC of 20,000 messages. Similar to the case for trial 3, 10,000 messages, we do not see any one factor contributing to the larger MaxEC.

The MaxEC of trial 4, 5,000 messages is the lowest of the five trials. We see in Table 6 that the MaxECN served only once as a CH, broadcast CH and sink node’s CH. This means that in each rotation of the CHs over the trial, a new node served as the sink node’s CH, and there are likely several other nodes in the WSN with energy consumption similar to the MaxECN.

In trial 4, 15,000 messages, the MaxEC is less than the average MaxEC of 10,000 messages. Examining the data in Table 6, we see that the roles performed by the MaxEC are similar to the other trials at this traffic volume. The number of nodes broadcast to is near the average of the five trials for 15,000 messages, so it is not contributing to the smaller MaxEC either. We see in Table 5 that the average energy expended for a node in this trial is also less than the other four trials and conclude this trial was simply more energy efficient than the other ones at this traffic volume.

The MaxEC of trial 5, 15,000 messages is also less than the average MaxEC of 10,000 messages. The MaxECN served as a CH five times but as a broadcast CH and the sink node’s CH only three times. This was the lowest of the five trials and contributed to the lower MaxEC.

#### *b. Sink Node Anonymity*

The anonymity factor is calculated based on the total number of nodes broadcast to as outlined in Chapter V, Section D. The results returned from Anony_Metrics are listed in Table 6 and used to calculate the anonymity factor for each trial and traffic volume.

The average number of nodes broadcast to ranges from 25.6000 to 31.5833 and exceeds the desired lower threshold of 20 nodes. There is a slight increase in the average number of nodes broadcast to with the increase in traffic volume, as shown in Figure 25. The slight increase in number of nodes broadcast to as traffic volume increases translates to a slight decrease in the anonymity factor as traffic volume increases is shown in Figure 26. By taking the average of all of the anonymity factors calculated in Table 6, we calculate to average anonymity factor of the topology to be 0.0357.

Table 6. The total number of nodes broadcast to and the anonymity factor for each trial and traffic volume of Topology 2.

| 5000 Messages | | | | | |
|---|---|---|---|---|---|
| Trial | 1 | 2 | 3 | 4 | 5 |
| Avg # of Nodes Broadcast To | 27.0000 | 26.2857 | 28.1667 | 25.6000 | 26.6667 |
| Anonymity Factor | 0.0370 | 0.0380 | 0.0355 | 0.0391 | 0.0375 |
| Avg # of Nodes in Sink Node Cluster Head | 7.0000 | 11.4286 | 7.5000 | 5.2000 | 6.3333 |
| 10,000 Messages | | | | | |
| Trial | 1 | 2 | 3 | 4 | 5 |
| Avg # of Nodes Broadcast To | 28.3636 | 27.1818 | 31.5833 | 27.1818 | 29.5455 |
| Anonymity Factor | 0.0353 | 0.0368 | 0.0317 | 0.0368 | 0.0338 |
| Avg # of Nodes in Sink Node Cluster Head | 7.3636 | 6.5455 | 9.2500 | 7.0000 | 8.4545 |
| 15,000 Messages | | | | | |
| Trial | 1 | 2 | 3 | 4 | 5 |
| Avg # of Nodes Broadcast To | 28.0588 | 29.2353 | 29.0588 | 27.5625 | 27.6875 |
| Anonymity Factor | 0.0356 | 0.0342 | 0.0344 | 0.0363 | 0.0361 |
| Avg # of Nodes in Sink Node Cluster Head | 7.8235 | 7.5882 | 8.2941 | 7.0000 | 7.3125 |
| 20,000 Messages | | | | | |
| Trial | 1 | 2 | 3 | 4 | 5 |
| Avg # of Nodes Broadcast To | 27.8095 | 29.8636 | 28.6818 | 28.0909 | 27.8095 |
| Anonymity Factor | 0.0360 | 0.0335 | 0.0349 | 0.0356 | 0.0360 |
| Avg # of Nodes in Sink Node Cluster Head | 5.9524 | 7.3182 | 8.1818 | 7.2727 | 6.9048 |
| Average Anonymity Factor of Topology | | | | | 0.0357 |

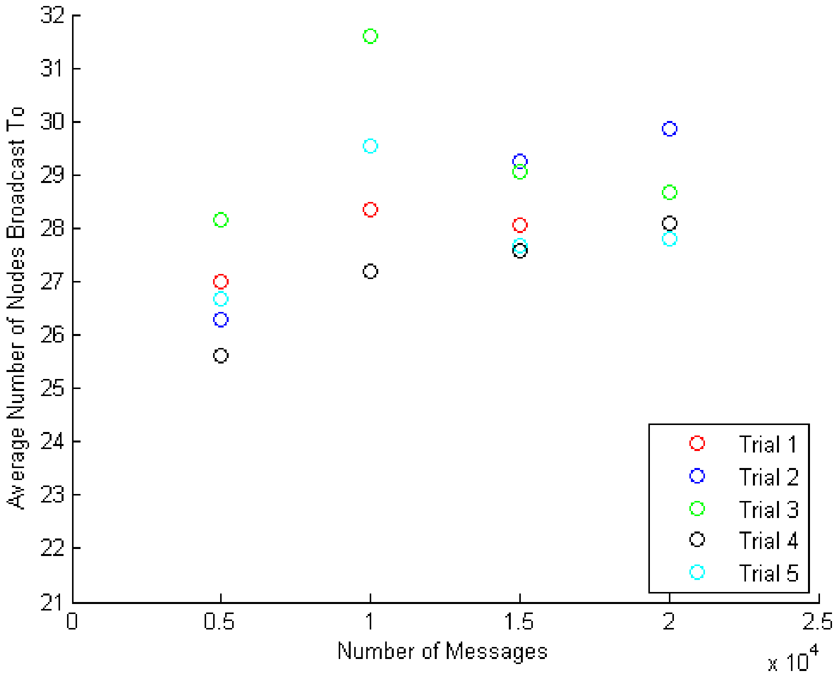


Figure 25. The average number of nodes broadcast to for different traffic volumes over the five trials. The average number of nodes broadcast to is between 25.6000 and 31.5833 for Topology 2.

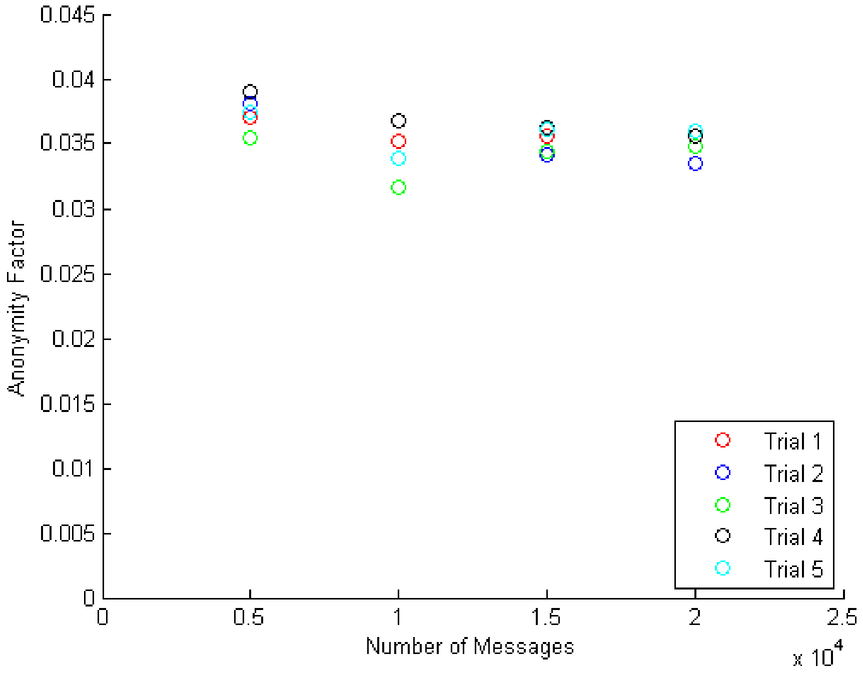


Figure 26. The anonymity factor of each trial at each traffic volume for Topology 2.

### 3. Topology 3

Just as in the previous topologies, the physical topology of Topology 3 was generated using the *Create_RandomSensorNetwork* discussed in Section A1 of this chapter. The physical location of the nodes remains the same throughout Topology 3. Across the five trials at each simulated traffic length, the only thing that changes is the role each nodes plays in the WSN.

#### *a. Energy Results*

The simulations described in Section A of this chapter are conducted over Topology 3, and the results are summarized in Table 7. This information is used to generate Figures 27, 28 and 29.

Table 7. The average, maximum, and minimum energy consumed by nodes in Topology 3 over five trials at each traffic volume. The roles played by the maximum energy node, contributed to an understanding of what drives the energy consumption of the MaxECN.

| | | | 5000 Messages | | |
|---|---|---|---|---|---|
| Trial | 1 | 2 | 3 | 4 | 5 |
| Energy | | | | | |
| Average Energy Expended by a Node | 4.1291x10$^{-4}$ | 4.3267x10$^{-4}$ | 4.2409x10$^{-4}$ | 4.2857x10$^{-4}$ | 4.1346x10$^{-4}$ |
| Min Energy Expended By a Node | 2.0150x10$^{-5}$ | 1.5450x10$^{-5}$ | 1.5350x10$^{-5}$ | 1.5450x10$^{-5}$ | 1.9050x10$^{-5}$ |
| Max Energy Expended By a Node | 9.3000x10$^{-3}$ | 8.2000x10$^{-3}$ | 7.2000x10$^{-3}$ | 5.1000x10$^{-3}$ | 8.2000x10$^{-3}$ |
| Distance of Max Energy Node from Sink Node | 11.2845 | 5.4562 | 5.4562 | 9.0608 | 5.4562 |
| # of Messages Max Energy Node Sends | 41 | 60 | 49 | 59 | 51 |
| # of times Max Energy Node is a Cluster Member | 3 | 4 | 3 | 5 | 4 |
| # of times Max Energy Node is a Cluster Head | 3 | 2 | 2 | 1 | 2 |
| # of times Max Energy Node is a Broadcast Cluster Head | 3 | 2 | 2 | 1 | 2 |
| # of times Max Energy Node is Sink Node Cluster Head | 3 | 2 | 2 | 1 | 2 |
| | | | 10,000 Messages | | |
| Trial | 1 | 2 | 3 | 4 | 5 |
| Energy | | | | | |
| Average Energy Expended by a Node | 8.9046x10$^{-4}$ | 8.3408x10$^{-4}$ | 8.1112x10$^{-4}$ | 7.6622x10$^{-4}$ | 9.2478x10$^{-4}$ |
| Min Energy Expended By a Node | 3.7000x10$^{-5}$ | 3.2200x10$^{-5}$ | 3.9750x10$^{-5}$ | 3.3550x10$^{-5}$ | 4.2350x10$^{-5}$ |
| Max Energy Expended By a Node | 1.3300x10$^{-2}$ | 1.3800x10$^{-2}$ | 1.4300x10$^{-2}$ | 1.1600x10$^{-2}$ | 1.0200x10$^{-2}$ |
| Distance of Max Energy Node from Sink Node | 8.5039 | 11.7691 | 8.5039 | 8.5039 | 8.5039 |
| # of Messages Max Energy Node Sends | 118 | 96 | 97 | 112 | 101 |
| # of times Max Energy Node is a Cluster Member | 8 | 8 | 6 | 5 | 9 |
| # of times Max Energy Node is a Cluster Head | 3 | 3 | 5 | 6 | 2 |
| # of times Max Energy Node is a Broadcast Cluster Head | 3 | 3 | 5 | 4 | 2 |
| # of times Max Energy Node is Sink Node Cluster Head | 3 | 3 | 4 | 4 | 2 |
| | | | 15,000 Messages | | |
| Trial | 1 | 2 | 3 | 4 | 5 |
| Energy | | | | | |
| Average Energy Expended by a Node | 1.2000x10$^{-3}$ | 1.3000x10$^{-3}$ | 1.2000x10$^{-3}$ | 1.2000x10$^{-3}$ | 1.2000x10$^{-3}$ |
| Min Energy Expended By a Node | 4.9650x10$^{-5}$ | 6.0850x10$^{-5}$ | 4.7800x10$^{-5}$ | 5.0250x10$^{-5}$ | 4.3900x10$^{-5}$ |
| Max Energy Expended By a Node | 1.8500x10$^{-2}$ | 1.9000x10$^{-2}$ | 2.3300x10$^{-2}$ | 1.7500x10$^{-2}$ | 2.0800x10$^{-2}$ |
| Distance of Max Energy Node from Sink Node | 8.5039 | 8.5039 | 8.5039 | 5.4562 | 5.4562 |
| # of Messages Max Energy Node Sends | 156 | 148 | 155 | 134 | 150 |
| # of times Max Energy Node is a Cluster Member | 8 | 11 | 8 | 12 | 10 |
| # of times Max Energy Node is a Cluster Head | 8 | 5 | 8 | 5 | 6 |
| # of times Max Energy Node is a Broadcast Cluster Head | 4 | 5 | 7 | 5 | 6 |
| # of times Max Energy Node is Sink Node Cluster Head | 4 | 5 | 7 | 5 | 6 |
| | | | 20,000 Messages | | |
| Trial | 1 | 2 | 3 | 4 | 5 |
| Energy | | | | | |
| Average Energy Expended by a Node | 1.6000x10$^{-3}$ | 1.7000x10$^{-3}$ | 1.6000x10$^{-3}$ | 1.6000x10$^{-3}$ | 1.5000x10$^{-3}$ |
| Min Energy Expended By a Node | 6.7450x10$^{-5}$ | 6.0050x10$^{-5}$ | 6.1550x10$^{-5}$ | 7.3800x10$^{-5}$ | 7.3200x10$^{-5}$ |
| Max Energy Expended By a Node | 1.500x10$^{-2}$ | 2.5100x10$^{-2}$ | 1.7600x10$^{-2}$ | 2.1500x10$^{-2}$ | 1.8400x10$^{-2}$ |
| Distance of Max Energy Node from Sink Node | 5.4562 | 7.1535 | 5.4562 | 8.5039 | 11.2845 |
| # of Messages Max Energy Node Sends | 201 | 176 | 215 | 181 | 185 |
| # of times Max Energy Node is a Cluster Member | 17 | 15 | 15 | 15 | 11 |
| # of times Max Energy Node is a Cluster Head | 4 | 6 | 6 | 6 | 10 |
| # of times Max Energy Node is a Broadcast Cluster Head | 4 | 6 | 6 | 6 | 6 |
| # of times Max Energy Node is Sink Node Cluster Head | 4 | 6 | 6 | 6 | 6 |

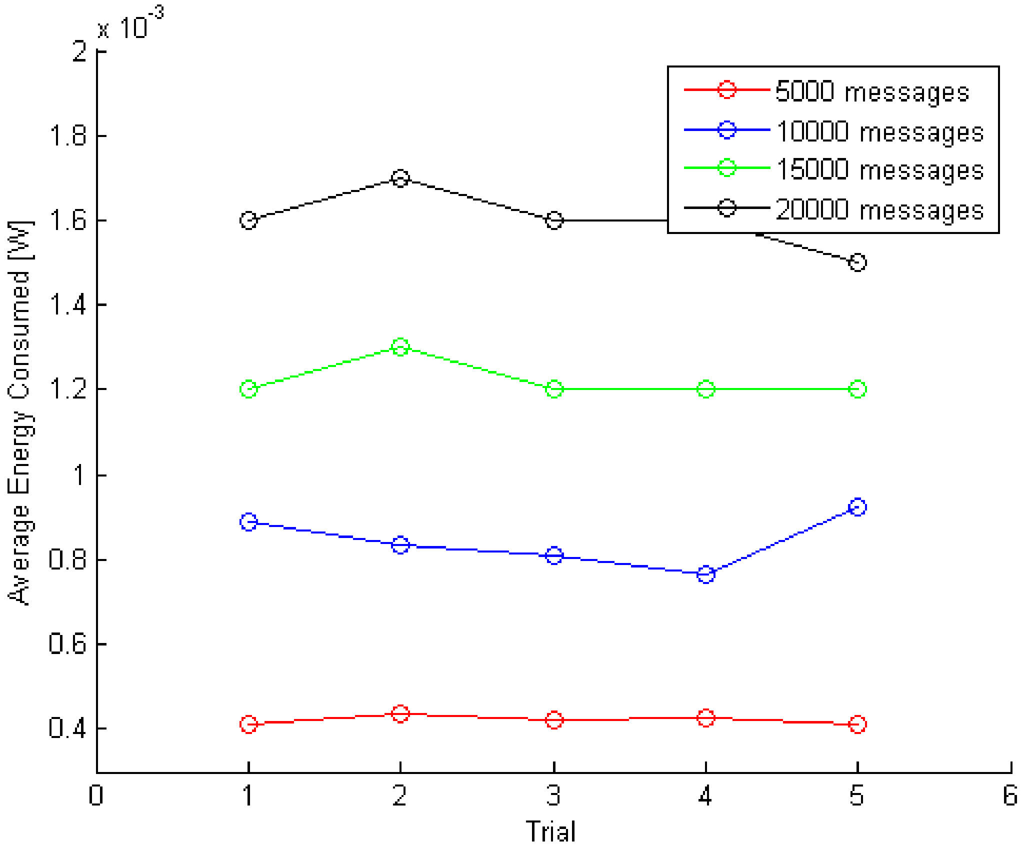


Figure 27. The average energy consumed by nodes in the WSN for 5,000, 10,000, 15,000, and 20,000 messages. The average energy consumed increases as traffic volume increases in all five trials in Topology 3.

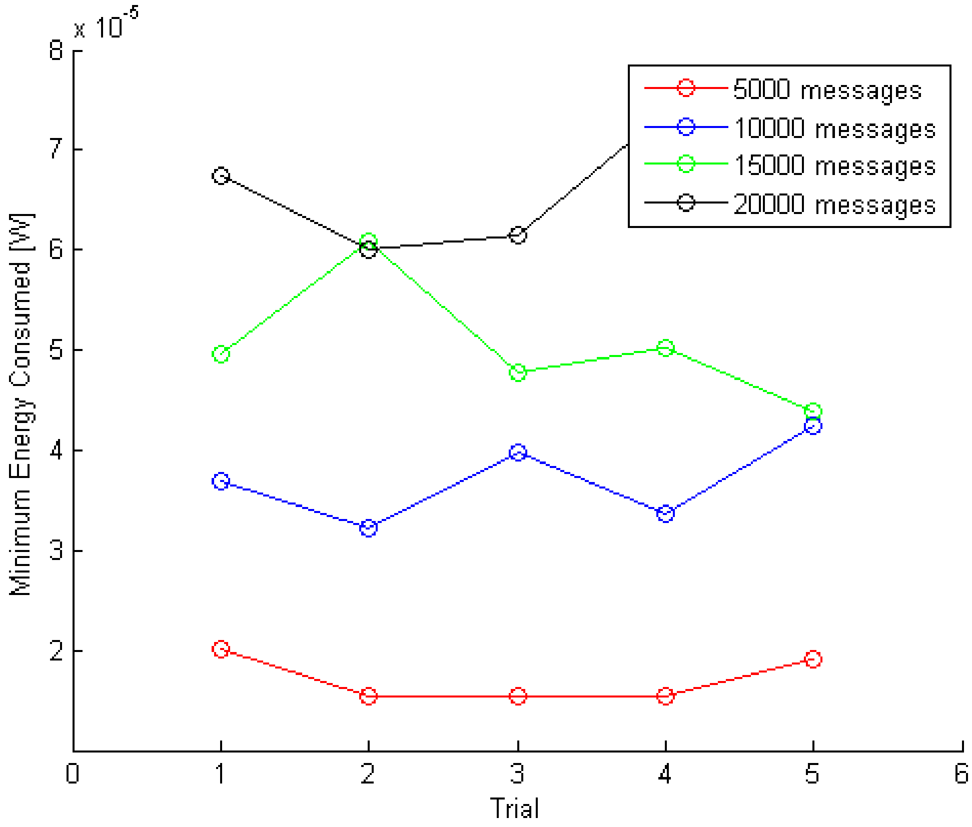


Figure 28. The minimum energy consumed by nodes in the WSN for 5,000, 10,000, 15,000, and 20,000 messages. The minimum energy consumed increases as traffic volume increases in all five trials in Topology 3. At trial 2, 15,000 messages, MinEC deviates from the average MinEC at that traffic volume.

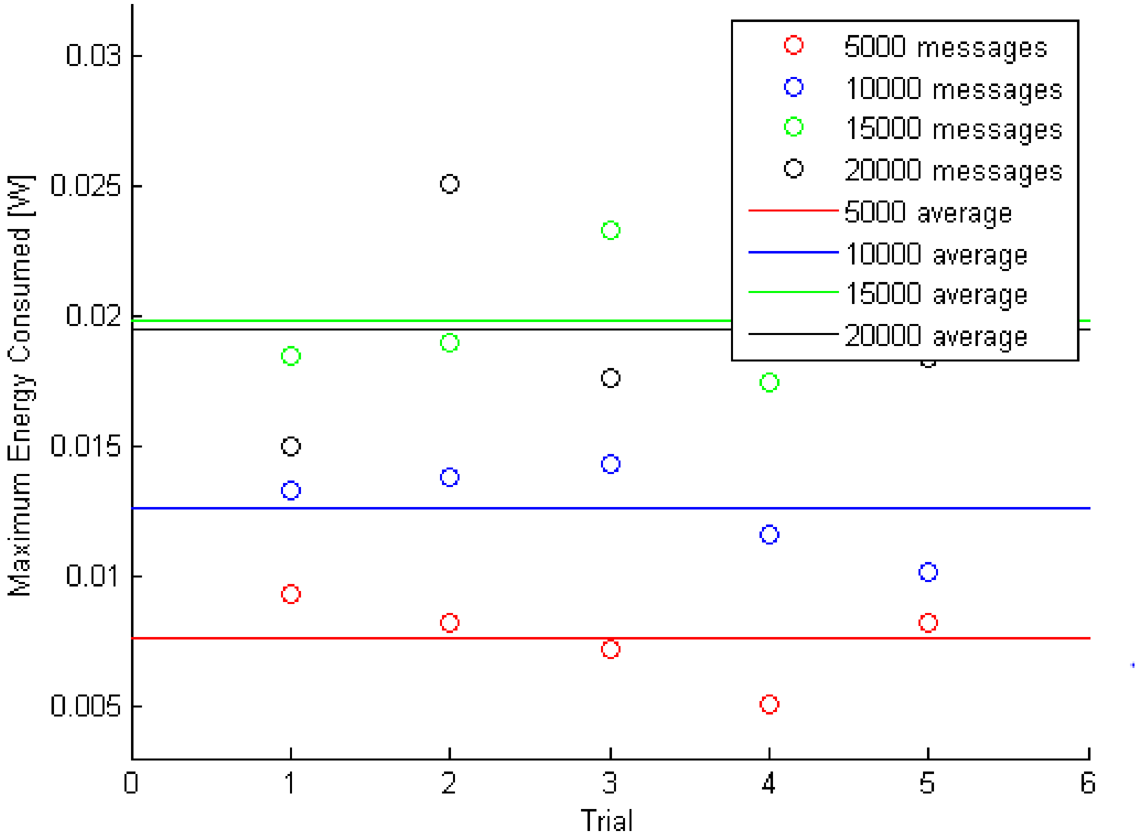


Figure 29. The maximum energy consumed by nodes in the WSN for 5,000, 10,000, 15,000, and 20,000 messages. The maximum energy consumed increases as traffic volume increases from 5,000 to 10,000 messages. The average maximum energy consumed for 15,000 messages slightly exceeds the average maximum energy consumed for 20,000 messages.

The AvgEC by a node in the WSN is consistent across the five trials and increases with the increase in traffic volume across the WSN, as shown in Figure 27. The MinEC also follows the same trend, increasing with traffic volume. When compared to Figure 27, we see that the values of MinEC, illustrated in Figure 28, have a larger deviation from the average MinEC of the five trials at each traffic volume.

The result of the Maximum Energy consumed by a node, shown in Figure 29, is very peculiar for this topology. The average MaxEC for 15,000 messages is slightly higher than the MaxEC for 20,000 messages. In trial 1, trial 3, and trial 5 for 15,000 messages, the MaxEC is less than the average MaxEC for 20,000 messages, but the overall average is still higher for 15,000 messages than 20,000 messages. The largest MaxEC of the two traffic volumes occurs at trial 2, 20,000 messages, but the smallest MaxEC of the two traffic volumes occurs at trial 1, 20,000 messages. If we remove trial 1, 20,000 messages from the data set, the order is restored. Examining Table 7, we see that trial 1, 20,000 messages is quite the anomaly only serving as a CH, broadcast CH and the sink node's CH four times compared to a minimum of six the other trials at this traffic volume. The average energy expended by a node for trial 1, 20,000 messages, is the same as the other trials, so we conclude that while this is an unexpected result this trial was simply more balanced in choosing multiple nodes to fill these functions over the simulation.

#### *b. Sink Node Anonymity*

The anonymity factor is calculated based on the total number of nodes broadcast to, as outlined in Chapter V, Section D. The results returned from *Anony_Metrics* are listed in Table 8 and are used to calculate the anonymity factor for each trial and traffic volume. The average number of nodes broadcast to ranges from 21.8125 to 29.1818 and exceeds the desired lower threshold of 20 nodes. The results vary based on traffic volume and do not demonstrate any trends of convergence to a number of numbers broadcast to or divergence from a number of nodes broadcast to as traffic volume increases, as shown in Figure 30. Just as there are no trends in the average number of nodes broadcast to, there are no trends on the anonymity factor over the different traffic volumes. This is shown in Figure 31. By taking the average of all of the anonymity factors calculated in Table 8, we calculate to average anonymity factor of the topology to be 0.0367.

Table 8. The total number of nodes broadcast to and the anonymity factor for each trial and traffic volume of Topology 3.

| | | | | | |
|---|---|---|---|---|---|
| 5000 Messages | | | | | |
| Trial | 1 | 2 | 3 | 4 | 5 |
| Avg # of Nodes Broadcast To | 28.5000 | 27.3333 | 28.6000 | 26.0000 | 25.3333 |
| Anonymity Factor | 0.0351 | 0.0366 | 0.0350 | 0.0385 | 0.0395 |
| Avg # of Nodes in Sink Node Cluster Head | 5.5000 | 8.5000 | 6.6000 | 7.0000 | 5.8333 |
| 10,000 Messages | | | | | |
| Trial | 1 | 2 | 3 | 4 | 5 |
| Avg # of Nodes Broadcast To | 27.2727 | 29.1818 | 27.4545 | 28.2727 | 27.7273 |
| Anonymity Factor | 0.0367 | 0.0343 | 0.0364 | 0.0354 | 0.0361 |
| Avg # of Nodes in Sink Node Cluster Head | 7.2727 | 6.1818 | 5.8182 | 5.9091 | 8.3636 |
| 15,000 Messages | | | | | |
| Trial | 1 | 2 | 3 | 4 | 5 |
| Avg # of Nodes Broadcast To | 28.0625 | 26.8750 | 21.8125 | 28.0000 | 27.1875 |
| Anonymity Factor | 0.0356 | 0.0372 | 0.0458 | 0.0357 | 0.0368 |
| Avg # of Nodes in Sink Node Cluster Head | 6.6250 | 7.1250 | 5.5625 | 6.4375 | 6.6250 |
| 20,000 Messages | | | | | |
| Trial | 1 | 2 | 3 | 4 | 5 |
| Avg # of Nodes Broadcast To | 27.3810 | 28.9524 | 27.6910 | 27.0952 | 26.4286 |
| Anonymity Factor | 0.0365 | 0.0345 | 0.0361 | 0.0369 | 0.0378 |
| Avg # of Nodes in Sink Node Cluster Head | 6.3333 | 7.3333 | 6.1429 | 6.2857 | 4.5238 |
| Average Anonymity Factor of Topology | | | | | 0.0367 |

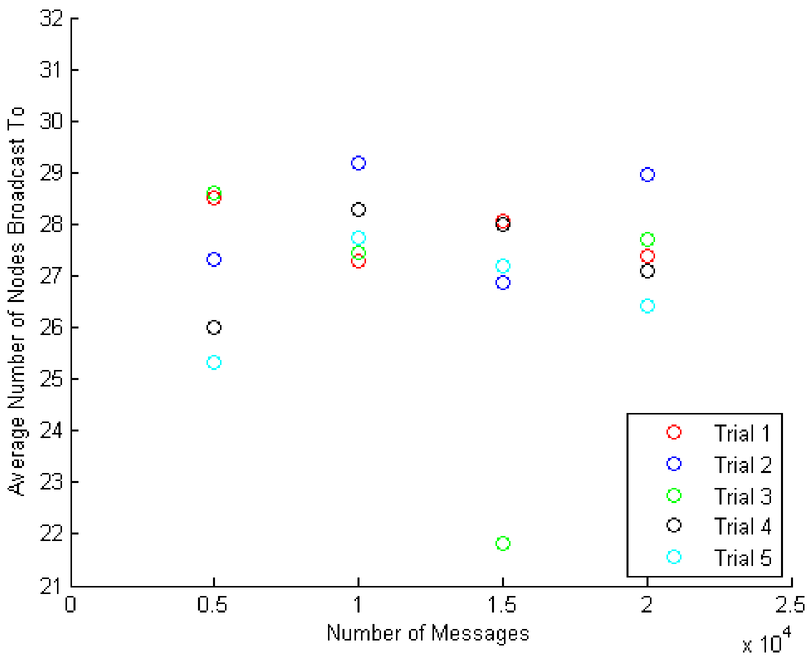


Figure 30. The average number of nodes broadcast to for different traffic volumes over five trials. The average number of nodes broadcast to is between 21.8125 and 29.1818 for Topology 3.

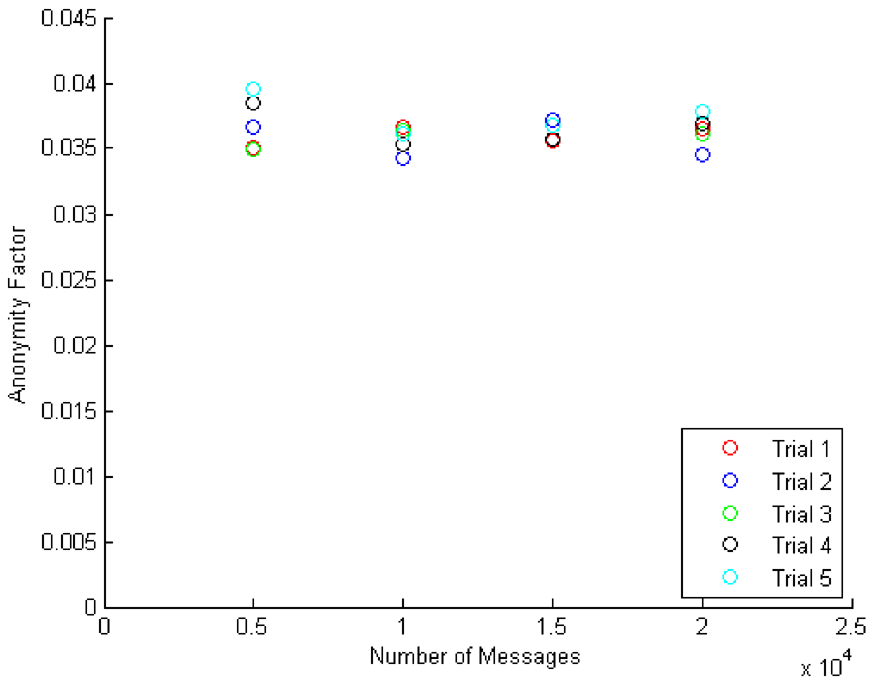


Figure 31. The anonymity factor of each trial at each traffic volume for Topology 3.

### 4. Topology 4

Just as in the previous topologies, the physical topology of Topology 4 was generated using the *Create_RandomSensorNetwork* discussed in Section A1 of this chapter. The physical location of the nodes remains the same throughout Topology 4. Across the five trials at each simulated traffic length, the only thing that changes is the role each nodes plays in the WSN.

#### *a. Energy Results*

The simulations described in Section A of this chapter are conducted over Topology 4, and the results are summarized in Table 9. This information is used to generate Figures 32, 33, and 34.

Table 9. The average, maximum, and minimum energy consumed by nodes in Topology 4 over five trials at each traffic volume. The roles played by the maximum energy node contributed to an understanding of what drives the energy consumption of the MaxECN.

| | 5000 Messages | | | | |
|---|---|---|---|---|---|
| Trial | 1 | 2 | 3 | 4 | 5 |
| Energy | | | | | |
| Average Energy Expended by a Node | $5.1530\text{x}10^{-4}$ | $4.6025\text{x}10^{-4}$ | $3.9675\text{x}10^{-4}$ | $4.3472\text{x}10^{-4}$ | $4.4179\text{x}10^{-4}$ |
| Min Energy Expended By a Node | $1.7850\text{x}10^{-5}$ | $1.3300\text{x}10^{-5}$ | $1.2600\text{x}10^{-5}$ | $1.6650\text{x}10^{-5}$ | $1.5800\text{x}10^{-5}$ |
| Max Energy Expended By a Node | $7.7000\text{x}10^{-3}$ | $1.0400\text{x}10^{-2}$ | $5.1000\text{x}10^{-3}$ | $5.1000\text{x}10^{-3}$ | $5.1000\text{x}10^{-3}$ |
| Distance of Max Energy Node from Sink Node | 3.3141 | 12.7595 | 16.0760 | 10.6008 | 16.0760 |
| # of Messages Max Energy Node Sends | 52 | 47 | 52 | 48 | 46 |
| # of times Max Energy Node is a Cluster Member | 4 | 3 | 5 | 5 | 4 |
| # of times Max Energy Node is a Cluster Head | 2 | 3 | 1 | 1 | 2 |
| # of times Max Energy Node is a Broadcast Cluster Head | 2 | 3 | 1 | 1 | 2 |
| # of times Max Energy Node is Sink Node Cluster Head | 2 | 3 | 1 | 1 | 2 |
| | 10,000 Messages | | | | |
| Trial | 1 | 2 | 3 | 4 | 5 |
| Energy | | | | | |
| Average Energy Expended by a Node | $8.3935\text{x}10^{-4}$ | $8.3360\text{x}10^{-4}$ | $8.7043\text{x}10^{-4}$ | $9.6332\text{x}10^{-4}$ | $8.4644\text{x}10^{-4}$ |
| Min Energy Expended By a Node | $2.8450\text{x}10^{-5}$ | $3.0600\text{x}10^{-5}$ | $2.9650\text{x}10^{-5}$ | $3.7244\text{x}10^{-5}$ | $3.1950\text{x}10^{-5}$ |
| Max Energy Expended By a Node | $1.1000\text{x}10^{-2}$ | $8.2000\text{x}10^{-3}$ | $1.1500\text{x}10^{-2}$ | $1.2300\text{x}10^{-2}$ | $1.0400\text{x}10^{-2}$ |
| Distance of Max Energy Node from Sink Node | 17.5268 | 17.2269 | 17.2269 | 10.6008 | 3.3141 |
| # of Messages Max Energy Node Sends | 115 | 89 | 92 | 101 | 88 |
| # of times Max Energy Node is a Cluster Member | 7 | 9 | 7 | 7 | 8 |
| # of times Max Energy Node is a Cluster Head | 4 | 2 | 4 | 4 | 3 |
| # of times Max Energy Node is a Broadcast Cluster Head | 2 | 2 | 3 | 4 | 2 |
| # of times Max Energy Node is Sink Node Cluster Head | 2 | 2 | 3 | 3 | 2 |
| | 15,000 Messages | | | | |
| Trial | 1 | 2 | 3 | 4 | 5 |
| Energy | | | | | |
| Average Energy Expended by a Node | $1.3000\text{x}10^{-3}$ | $1.2000\text{x}10^{-3}$ | $1.3000\text{x}10^{-3}$ | $1.2000\text{x}10^{-3}$ | $1.1000\text{x}10^{-3}$ |
| Min Energy Expended By a Node | $4.6550\text{x}10^{-5}$ | $4.8050\text{x}10^{-5}$ | $4.7300\text{x}10^{-5}$ | $4.8200\text{x}10^{-5}$ | $5.3700\text{x}10^{-5}$ |
| Max Energy Expended By a Node | $1.9100\text{x}10^{-2}$ | $1.3100\text{x}10^{-2}$ | $1.6000\text{x}10^{-2}$ | $7.9000\text{x}10^{-3}$ | $2.4200\text{x}10^{-2}$ |
| Distance of Max Energy Node from Sink Node | 2.8057 | 13.7577 | 3.3141 | 3.3141 | 2.8057 |
| # of Messages Max Energy Node Sends | 134 | 152 | 155 | 151 | 160 |
| # of times Max Energy Node is a Cluster Member | 10 | 11 | 10 | 12 | 7 |
| # of times Max Energy Node is a Cluster Head | 6 | 5 | 7 | 4 | 9 |
| # of times Max Energy Node is a Broadcast Cluster Head | 6 | 4 | 5 | 3 | 9 |
| # of times Max Energy Node is Sink Node Cluster Head | 6 | 3 | 5 | 3 | 9 |
| | 20,000 Messages | | | | |
| Trial | 1 | 2 | 3 | 4 | 5 |
| Energy | | | | | |
| Average Energy Expended by a Node | $1.7000\text{x}10^{-3}$ | $1.7000\text{x}10^{-3}$ | $1.7000\text{x}10^{-3}$ | $1.6000\text{x}10^{-3}$ | $1.8000\text{x}10^{-3}$ |
| Min Energy Expended By a Node | $5.3350\text{x}10^{-5}$ | $6.1900\text{x}10^{-5}$ | $5.7700\text{x}10^{-5}$ | $5.9100\text{x}10^{-5}$ | $7.6900\text{x}10^{-5}$ |
| Max Energy Expended By a Node | $2.8600\text{x}10^{-2}$ | $1.3700\text{x}10^{-2}$ | $3.0900\text{x}10^{-2}$ | $1.6900\text{x}10^{-2}$ | $2.0100\text{x}10^{-2}$ |
| Distance of Max Energy Node from Sink Node | 10.6008 | 3.3141 | 2.8057 | 2.8057 | 2.8057 |
| # of Messages Max Energy Node Sends | 201 | 178 | 188 | 179 | 193 |
| # of times Max Energy Node is a Cluster Member | 14 | 14 | 12 | 16 | 15 |
| # of times Max Energy Node is a Cluster Head | 7 | 7 | 10 | 5 | 6 |
| # of times Max Energy Node is a Broadcast Cluster Head | 7 | 3 | 10 | 5 | 6 |
| # of times Max Energy Node is Sink Node Cluster Head | 7 | 3 | 10 | 5 | 6 |

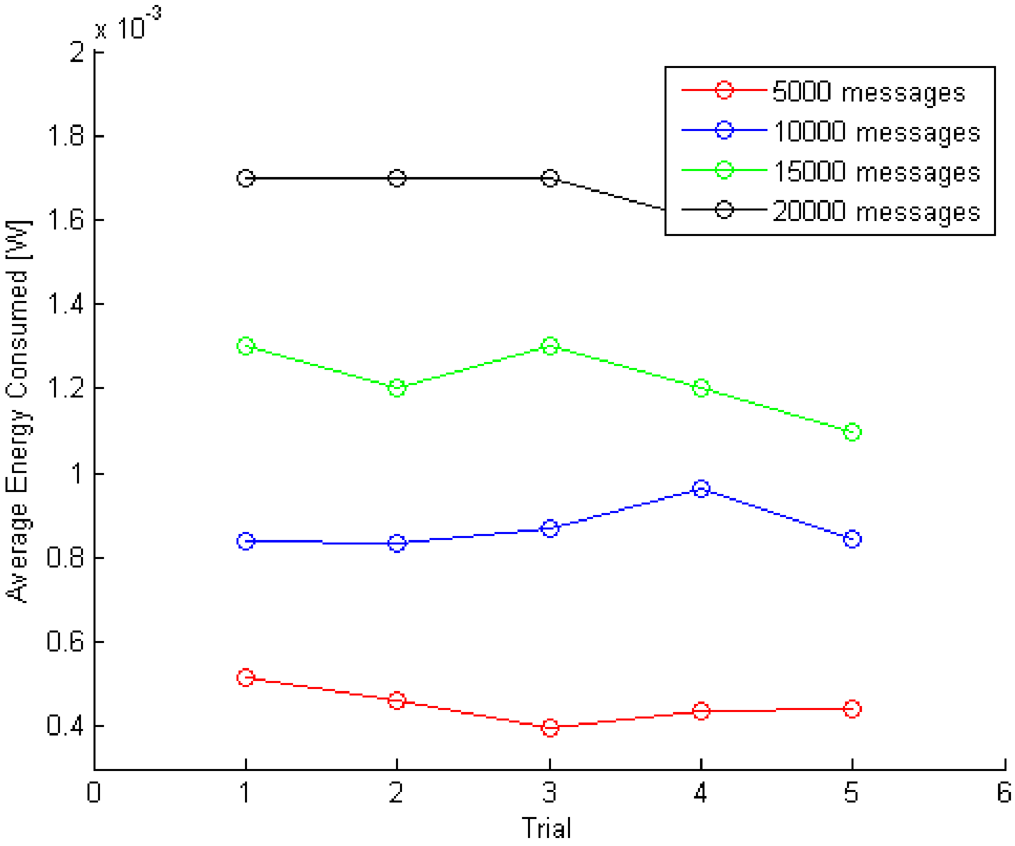


Figure 32. The average energy consumed by nodes in the WSN for 5,000, 10,000, 15,000, and 20,000 messages. The average energy consumed increases as traffic volume increases in all five trials.

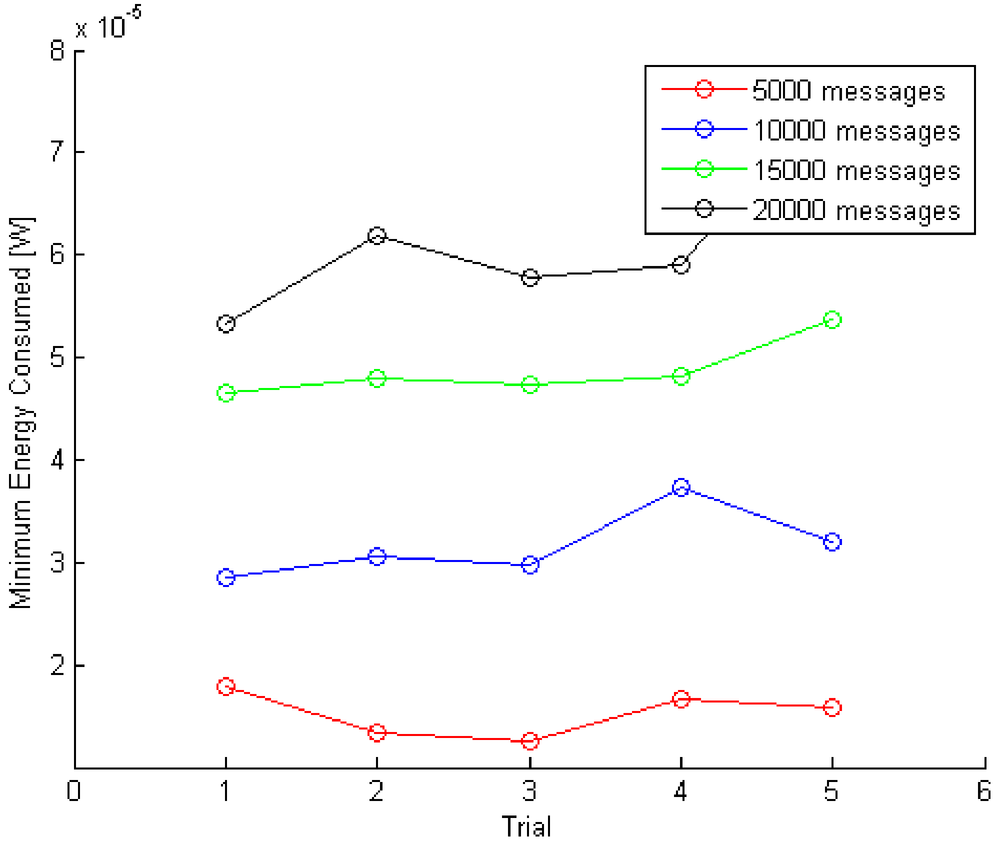


Figure 33. The minimum energy consumed by nodes in the WSN for 5,000, 10,000, 15,000, and 20,000 messages. The minimum energy consumed increases as traffic volume increases in all five trials in Topology 4.

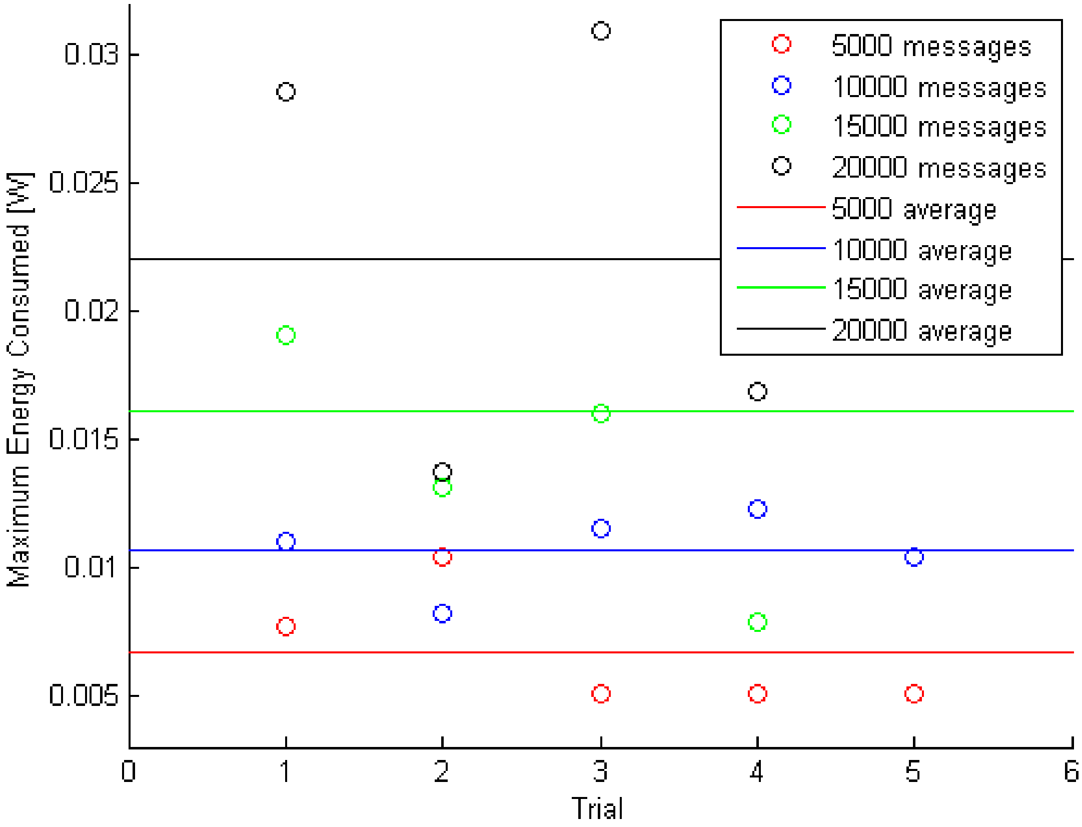


Figure 34. The maximum energy consumed by nodes in the WSN for 5,000, 10,000, 15,000 and 20,000 messages. The average MaxEC increases as traffic volume increases. However, the values of MaxEC vary dramatically across the trials. The MaxECN for 15,000 and 20,000 messages shows the greatest variation of the four topologies.

The AvgEC by a node in the WSN is consistent across the five trials and increases with the increase in traffic volume across the WSN, as shown in Figure 32. The MinEC also follows the same trend, increasing with the increase in traffic volume. When compared to Figure 32, we see that the values of MinEC, illustrated in Figure 33, have a larger deviation from the average MinEC of the five trials at each traffic volume.

In trial 2, 20,000 messages, the MaxEC is less than the average MaxEC of 15,000 messages, as shown in Figure 34. We see in Table 9 that the MaxECN served as a broadcast CH and sink node’s CH only three times, compared to five or more for the other four trials. The average energy expended by a node for trial 2, 20,000 messages is the same as the other trials. Thus, we conclude that this trial was simply more balanced in choosing multiple nodes to fill these functions over the simulation lowering the MaxEC. This is similar to what we observed in Topology 3.

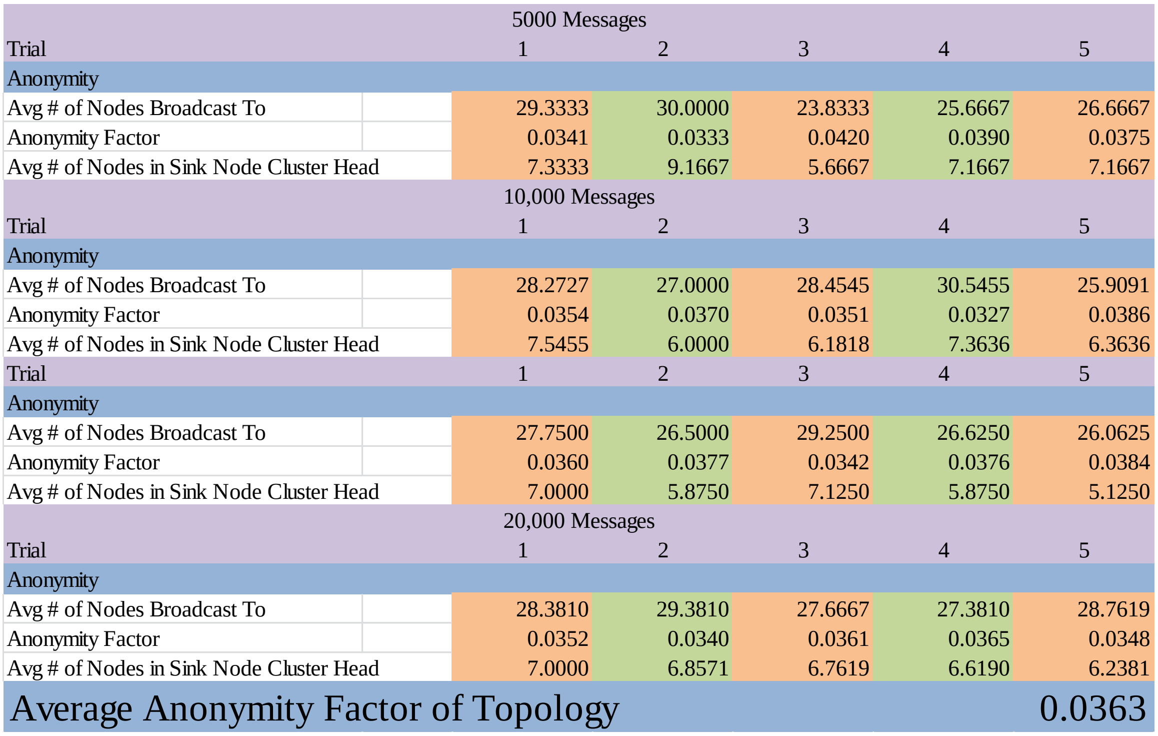
The MaxEC of trial 4, 15,000 messages is less than the average MaxEC of 10,000 messages. The MaxECN performs the roles of cluster member, CH, broadcast CH, and sink node's CH with similar frequency to all other MaxECN at this traffic volume, and the average number of nodes broadcast to is near the average of this traffic volume. There is no one factor that can be isolated to determine when the MaxEC of this trial is less than the average MaxEC of 10,000 messages. We simply conclude that the network initialization was more efficient.

In trial 5, 15,000 messages, the MaxEC is greater than the average MaxEC of 20,000 messages. The MaxECN serves as a cluster member only seven times compared to a minimum of ten in the other four trials at this traffic volume. The MaxECN serves at a CH, broadcast CH, and the sink node's CH nine times. This greatly exceeds the role played by any other MaxECN in the trials at this traffic volume and in four out of the five trials at the 20,000 message volume. We conclude that while the traffic was simulated in trial 5, 15,000 messages, the MaxECN was selected more than half the time to serve in roles that consume more energy. This drives the MaxEC for this node well above the average for 20,000 messages.

#### ***b. Sink Node Anonymity***

The anonymity factor is calculated based on the total number of nodes broadcast to, as outlined in Chapter V, Section D. The results returned from *Anony_Metrics* are listed in Table 10 and used to calculate the anonymity factor for each trial and traffic volume. The average number of nodes broadcast to ranges from 23.8333 to 30.0000 and exceeds the desired lower threshold of 20 nodes. The results vary based on traffic volume. In Figure 35, there is a slight convergence in the average number of nodes broadcast to as traffic volume increases. This translates in the same way to the anonymity factor. The anonymity factor values become more tightly grouped as traffic volume increases, as shown in Figure 36. By taking the average of all of the anonymity factors calculated in Table 10, we calculate to average anonymity factor of the topology to be 0.0363.

Table 10. The total number of nodes broadcast to and the anonymity factor for each trial and traffic volume of Topology 4.



| | 5000 Messages | | | | |
|---|---|---|---|---|---|
| Trial | 1 | 2 | 3 | 4 | 5 |
| Anonymity | | | | | |
| Avg # of Nodes Broadcast To | 29.3333 | 30.0000 | 23.8333 | 25.6667 | 26.6667 |
| Anonymity Factor | 0.0341 | 0.0333 | 0.0420 | 0.0390 | 0.0375 |
| Avg # of Nodes in Sink Node Cluster Head | 7.3333 | 9.1667 | 5.6667 | 7.1667 | 7.1667 |
| | 10,000 Messages | | | | |
| Trial | 1 | 2 | 3 | 4 | 5 |
| Anonymity | | | | | |
| Avg # of Nodes Broadcast To | 28.2727 | 27.0000 | 28.4545 | 30.5455 | 25.9091 |
| Anonymity Factor | 0.0354 | 0.0370 | 0.0351 | 0.0327 | 0.0386 |
| Avg # of Nodes in Sink Node Cluster Head | 7.5455 | 6.0000 | 6.1818 | 7.3636 | 6.3636 |
| Trial | 1 | 2 | 3 | 4 | 5 |
| Anonymity | | | | | |
| Avg # of Nodes Broadcast To | 27.7500 | 26.5000 | 29.2500 | 26.6250 | 26.0625 |
| Anonymity Factor | 0.0360 | 0.0377 | 0.0342 | 0.0376 | 0.0384 |
| Avg # of Nodes in Sink Node Cluster Head | 7.0000 | 5.8750 | 7.1250 | 5.8750 | 5.1250 |
| | 20,000 Messages | | | | |
| Trial | 1 | 2 | 3 | 4 | 5 |
| Anonymity | | | | | |
| Avg # of Nodes Broadcast To | 28.3810 | 29.3810 | 27.6667 | 27.3810 | 28.7619 |
| Anonymity Factor | 0.0352 | 0.0340 | 0.0361 | 0.0365 | 0.0348 |
| Avg # of Nodes in Sink Node Cluster Head | 7.0000 | 6.8571 | 6.7619 | 6.6190 | 6.2381 |
| Average Anonymity Factor of Topology | | | | | 0.0363 |

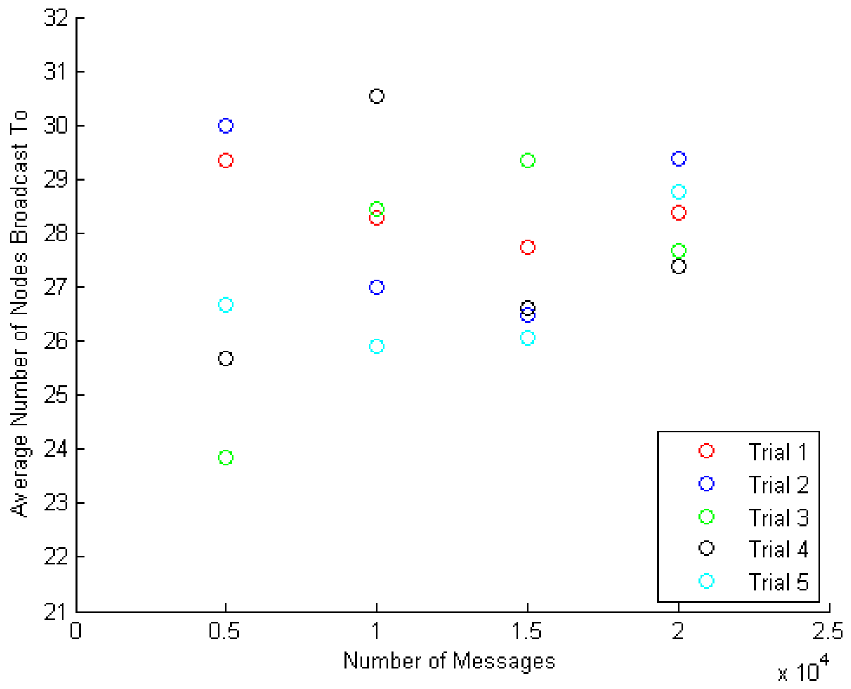


Figure 35. The average number of nodes broadcast to for different traffic volumes over five trials. The average number of nodes broadcast to is between 23.8333 and 30.0000 for Topology 4.

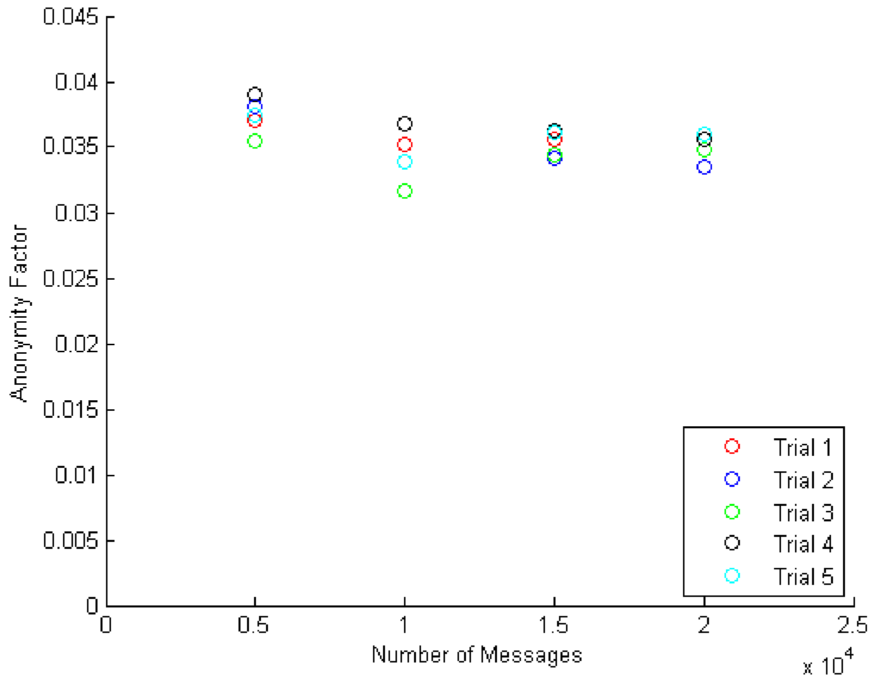


Figure 36. The anonymity factor of each trial at each traffic volume for Topology 4.

## 5. Energy Efficiency Conclusions

We have discussed the nuances of each topology in the previous sections and now examine the energy efficiency of the algorithm across the four topologies. These are all summarized and contained within Table 11.

Table 11. The Average, Maximum, and Minimum Energy Consumed by a node in the WSN across the four simulated topologies.

| | Topology 1 | Topology 2 | Topology 3 | Topology 4 | | Average |
|---|---|---|---|---|---|---|
| Avg Energy | | | | | | |
| 5000 Messages | $3.7475x10^{-4}$ | $4.2422x10^{-4}$ | $4.2306x10^{-4}$ | $4.4976x10^{-4}$ | | $4.1795x10^{-4}$ |
| 10000 Messages | $7.4586x10^{-4}$ | $8.3623x10^{-4}$ | $8.4533x10^{-4}$ | $8.7063x10^{-4}$ | | $8.2451x10^{-4}$ |
| 15000 Messages | $1.1400x10^{-3}$ | $1.2800x10^{-3}$ | $1.2200x10^{-3}$ | $1.2200x10^{-3}$ | | $1.2150x10^{-3}$ |
| 20000 Messages | $1.5000x10^{-3}$ | $1.6200x10^{-3}$ | $1.6000x10^{-3}$ | $1.7000x10^{-3}$ | | $1.6050x10^{-3}$ |
| Min Energy | | | | | | |
| 5000 Messages | $1.7940x10^{-5}$ | $1.7300x10^{-5}$ | $1.7090x10^{-5}$ | $1.5240x10^{-5}$ | | $1.6893x10^{-5}$ |
| 10000 Messages | $3.7630x10^{-5}$ | $3.0090x10^{-5}$ | $3.6970x10^{-5}$ | $3.0163x10^{-5}$ | | $3.3713x10^{-5}$ |
| 15000 Messages | $4.8850x10^{-5}$ | $4.3470x10^{-5}$ | $5.0490x10^{-5}$ | $4.8760x10^{-5}$ | | $4.7893x10^{-5}$ |
| 20000 Messages | $6.3120x10^{-5}$ | $5.7430x10^{-5}$ | $6.7210x10^{-5}$ | $6.1790x10^{-5}$ | | $6.2388x10^{-5}$ |
| Max Energy | | | | | | |
| 5000 Messages | $5.4000x10^{-3}$ | $9.4400x10^{-3}$ | $7.6000x10^{-3}$ | $6.6800x10^{-3}$ | | $7.2800x10^{-3}$ |
| 10000 Messages | $9.5200x10^{-3}$ | $1.4560x10^{-2}$ | $1.2640x10^{-2}$ | $1.0680x10^{-2}$ | | $1.1850x10^{-2}$ |
| 15000 Messages | $1.2080x10^{-2}$ | $1.6400x10^{-2}$ | $1.9820x10^{-2}$ | $1.6060x10^{-2}$ | | $1.6090x10^{-2}$ |
| 20000 Messages | $1.3780x10^{-2}$ | $2.0020x10^{-2}$ | $1.9520x10^{-2}$ | $2.2040x10^{-2}$ | | $1.8840x10^{-2}$ |

The average energy consumed by a node at each traffic volume is plotted for each topology in Figure 37. We see that the results are very similar to one another. The average energy consumed by a node increases as the traffic volume increases for each topology. Comparing the results side by side on the same plot, we see that the average energy consumed by a node in Topology 1 is consistently less than the other topologies. We expect the variation among the topologies as the physical location of the nodes will affect the energy consumption of each node in the WSN.

In several trials of the individual topologies, we could not isolate a single factor driving the values for the trial away from the average. For example, for trial 2, 15,000 messages of Topology 1, we conclude that the network initialization was more efficient. In other cases we, describe the network initialization as being less efficient. The source of this comes from *CH_isotest* discussed in Chapter VI, Section A2. If the topology fails the isolation test and must re-elect the cluster heads, this imposes an additional energy cost on the WSN, increasing the AvgEC, MinEC, and MaxEC.

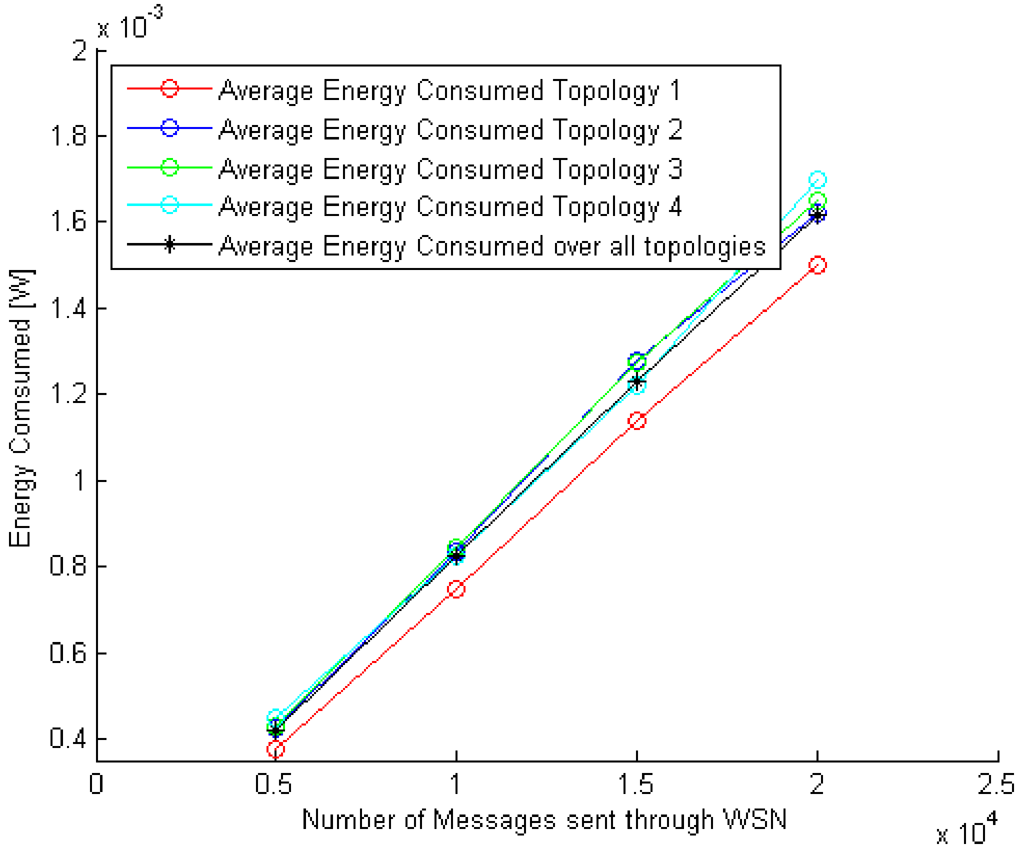


Figure 37. The average energy consumed by a node for all four topologies and the average of the four. The average energy consumed by a node in the WSN increases as the traffic volume through the WSN increases from 5,000 messages to 20,000 messages. The results are consistent across the four topologies simulated.

For each topology we examine the average minimum energy consumed by a node at each traffic volume. From Figure 38, we see that Topology 2 consistently has the minimum energy consumed by a node. The overall trend is that the consumption increases with traffic volume across all four topologies.  Similar to the average energy consumed, there is some variation among the results for the four topologies; however, the results are very consistent, with no points being large outliers.

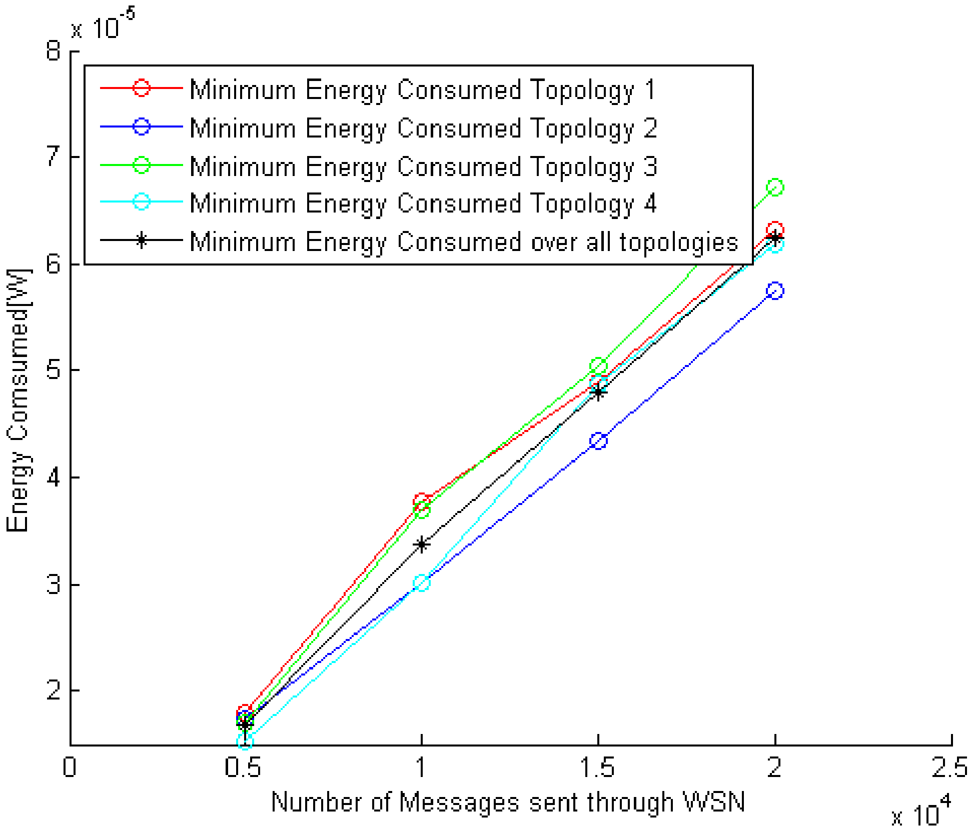


Figure 38. The minimum energy values for all four topologies and their average. The minimum energy consumed by a node in the WSN increases as the traffic volume through the WSN increases from 5,000 messages to 20,000 messages.

The maximum energy consumed by a node for each topology and traffic volume varies more than the average energy consumed and minimum energy consumed. These results are shown in Figure 39. From Figure 39 we see that the maximum energy consumed by a node are not as tightly grouped at any of the traffic volumes as they were in Figure 37 and Figure 38. Much like in the individual case for each trial in each topology, the maximum energy consumed is harder to predict because so many of the roles are chosen randomly, creating more variation.

Considering all four topologies individually and averaged together, we find remarkably consistent results for the average amount of energy consumed by a node in the WSN. This is promising because the average energy use by each node is an effective parameter for planning overall network lifetime. For simplicity, we did not let any nodes die out in these simulations because when nodes die the WSN may become partitioned, making the problem more difficult. Our goal was to evaluate the performance of the algorithm over a network where all of the nodes were alive.

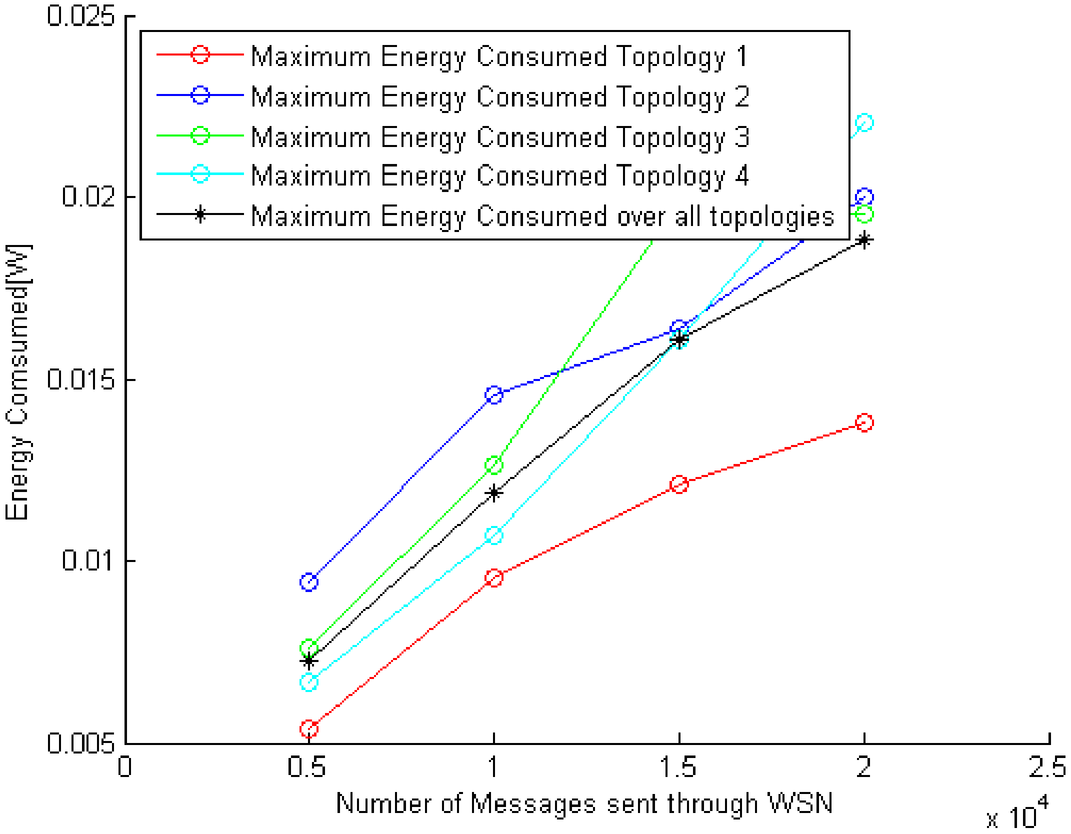


Figure 39. The maximum energy values for all four topologies and their average. The maximum energy consumed by a node in the WSN increases as the traffic volume through the WSN increase from 5,000 messages to 20,000 messages. Topology 1 consistently consumes less energy than Topologies 2, 3 and 4, but follows the same general trend of increased consumption with increased traffic volume.

### 6. Sink Node Anonymity Conclusions

The sink node anonymity was introduced in Chapter V, Section D in Eq. (9). We use this to evaluate the results of the simulations.

By taking the average of the five trials at each traffic volume for each topology, we eliminate the outliers and see that the average number of nodes broadcast to falls between 25.6724 and 28.7712 for all of the topologies. This is a much smaller range than we present in any one of the individual topologies. For the traffic volume of 5,000 messages, the average number of nodes broadcast to and the anonymity factor is tightly grouped. At the traffic volume of 10,000 messages, the highest and lowest number average number of nodes broadcast to are both present. At 15,000 and 20,000 messages, the range that the average values are spread over decreases again. These results are shown in Table 12.

We saw in Topology 1 and Topology 3 that the number of numbers broadcast to was independent of the traffic volume; thus, we conclude that the overall anonymity factor of any WSN is also independent, as illustrated in Figure 41. This is an important conclusion because, if the anonymity factor was reliant on a certain traffic volume, this would be a constraint for the employment of the algorithm and our objective is to have broad applications.

This conclusion leads us to examine the average number of nodes broadcast to at all message volumes, listed in Table 12, to determine the anonymity factor for the topology. We see very consistent results across the four topologies, as illustrated in Figure 42 and Figure 43. We see variation between the four topologies, just as we did in the energy efficiency conclusions, because some topologies may inherently lend themselves privacy preservation schemes. We also see that the results are remarkably consistent. The value of the anonymity factor for each topology is under 0.04. This means that for any given topology we simulated an adversary conducting traffic analysis of the deployed WSN would have a less than 4% chance of finding the sink node on his/her first guess when physically searching for the sensor.

Table 12. The results of the anonymity metrics is the average number of nodes broadcast to across the four topologies and four traffic volumes. This is used to determine the anonymity factor of the topologies.

| | Topology 1 | Topology 2 | Topology 3 | Topology 4 |
|---|---|---|---|---|
| Average Number of Total Number of Nodes Broadcast to by algorithm | | | | |
| 5000 Messages | 26.863320 | 26.743820 | 27.153320 | 27.100000 |
| 10000 Messages | 25.672740 | 28.771200 | 27.981800 | 28.036360 |
| 15000 Messages | 26.694700 | 28.320580 | 26.387500 | 27.237500 |
| 20000 Messages | 26.542860 | 28.451060 | 27.509640 | 28.314320 |
| Topology | 26.443405 | 28.071665 | 27.258065 | 27.672045 |
| Anonymity Factor | | | | |
| 5000 Messages | 0.037225 | 0.037392 | 0.036828 | 0.036900 |
| 10000 Messages | 0.038952 | 0.034757 | 0.035738 | 0.035668 |
| 15000 Messages | 0.037461 | 0.035310 | 0.037897 | 0.036714 |
| 20000 Messages | 0.037675 | 0.035148 | 0.036351 | 0.035318 |
| Topology | 0.037817 | 0.035623 | 0.036686 | 0.036138 |
| Average Anonymity Factor Across All Topologies | | | | 0.036566 |

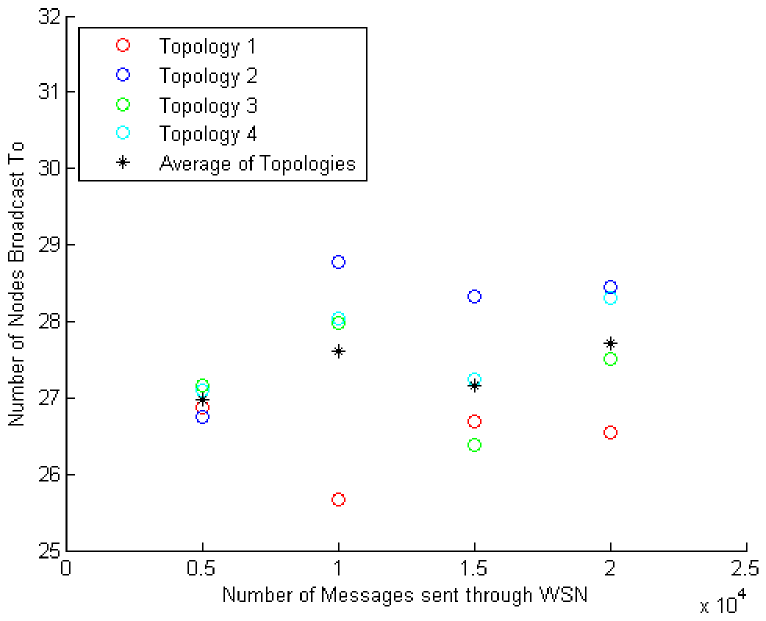


Figure 40. The average number of nodes broadcast for all four topologies and their average at each traffic volume. The number of nodes broadcast to varies over each topology and for each traffic volume but stays well above the established lower threshold of 20 nodes and in a relatively tight group of 25 to 30 nodes.

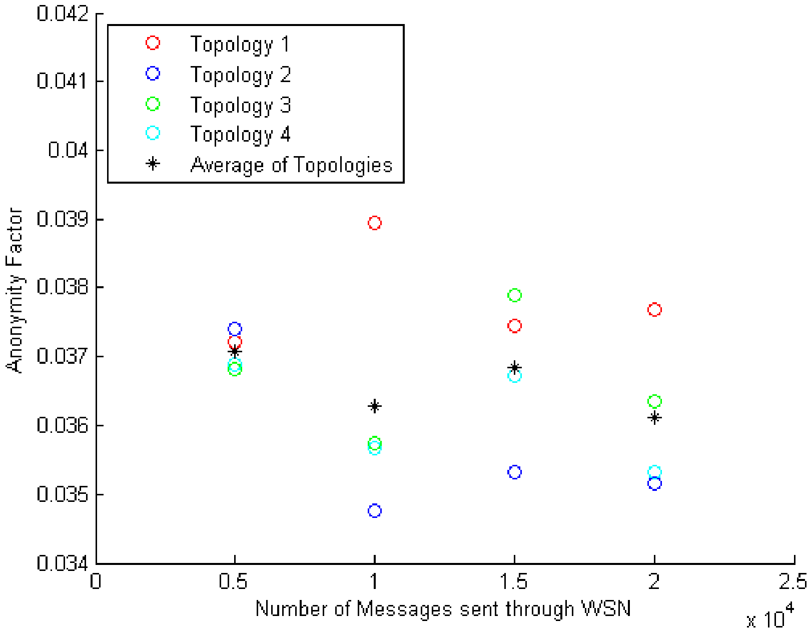


Figure 41. The anonymity factor of each topology and the average anonymity factor calculated at each traffic volume.

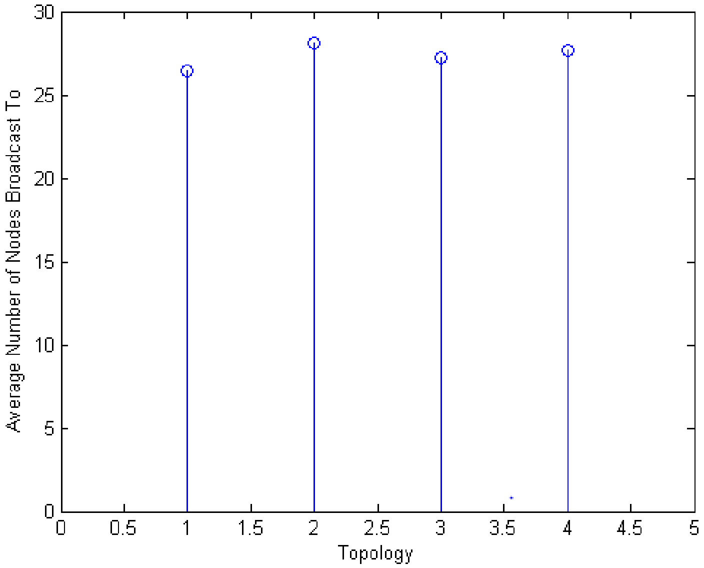


Figure 42. The average number of nodes broadcast to for each topology, with all traffic volumes included. After concluding that the anonymity is independent of traffic volume we consider all of the data points for number of nodes broadcast to for each topology to determine the average over the topology.

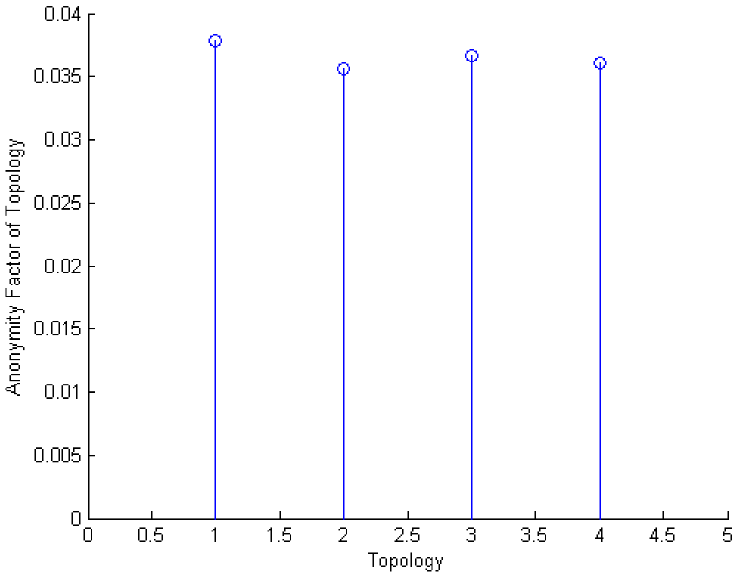


Figure 43. The anonymity factor for all four topologies. The anonymity factor of the topologies is calculated based on the average number of nodes broadcast to across all traffic volumes. The results are consistent across the four topologies.

## C. POTENTIAL ALGORITHM TRADEOFF

The energy efficiency and anonymity metrics reflect that the proposed algorithm meets the desired result of being a privacy preserving algorithm approach and is also mindful of the of overall energy consumption and network lifetime.

### 1. Additional Overhead from Broadcasting

The use of broadcast CHs generates extra network traffic. We see from Table 13 and Figure 44 that approximately 20 additional nodes receive broadcast traffic when compared to only the sink node's CH broadcasting, regardless of the topology. The extra traffic is necessary to achieve our desired level of anonymity, and the return on this extra traffic is significant with a more than 10% increase in the anonymity factor of the sink node as shown in Figure 45. While the excess traffic can be considered a tradeoff for higher anonymity, it should be noted that many of the schemes for privacy preservation in WSN introduced in Chapter II also have has significant overhead but do not achieve privacy preservation and energy efficiency simultaneously.

Table 13. A comparison of the average number of nodes broadcast to with and without the use of broadcast CHs. The use of broadcast CHs increase the number of nodes broadcast to and significantly reduces the anonymity factor of the sink node in the WSN.

| | Topology 1 | Topology 2 | Topology 3 | Topology 4 |
|---|---|---|---|---|
| Average Number of Total Number of Nodes Broadcast to by the algorithm | | | | |
| 5000 Messages | 26.863320 | 26.743820 | 27.153320 | 27.100000 |
| 10000 Messages | 25.672740 | 28.771200 | 27.981800 | 28.036360 |
| 15000 Messages | 26.694700 | 28.320580 | 26.387500 | 27.237500 |
| 20000 Messages | 26.542860 | 28.451060 | 27.509640 | 28.314320 |
| Topology | 26.443405 | 28.071665 | 27.258065 | 27.672045 |
| Anonymity Factor | | | | |
| 5000 Messages | 0.037225 | 0.037392 | 0.036828 | 0.036900 |
| 10000 Messages | 0.038952 | 0.034757 | 0.035738 | 0.035668 |
| 15000 Messages | 0.037461 | 0.035310 | 0.037897 | 0.036714 |
| 20000 Messages | 0.037675 | 0.035148 | 0.036351 | 0.035318 |
| Topology | 0.037817 | 0.035623 | 0.036686 | 0.036138 |
| Average Number of Sink Node Cluster Head Members | | | | |
| 5000 Messages | 5.092660 | 7.492380 | 6.686660 | 7.300020 |
| 10000 Messages | 4.218160 | 7.722720 | 6.709080 | 6.690900 |
| 15000 Messages | 5.094900 | 7.603660 | 6.475000 | 6.200000 |
| 20000 Messages | 4.933340 | 7.125980 | 6.123800 | 6.695220 |
| Topology | 4.834765 | 7.486185 | 6.498635 | 6.721535 |
| Anonymity Factor without Broadcast cluster heads | | | | |
| 5000 Messages | 0.196361 | 0.133469 | 0.149551 | 0.136986 |
| 10000 Messages | 0.237070 | 0.129488 | 0.149052 | 0.149457 |
| 15000 Messages | 0.196275 | 0.131516 | 0.154440 | 0.161290 |
| 20000 Messages | 0.202702 | 0.140332 | 0.163297 | 0.149360 |
| Topology | 0.206835 | 0.133579 | 0.153878 | 0.148776 |

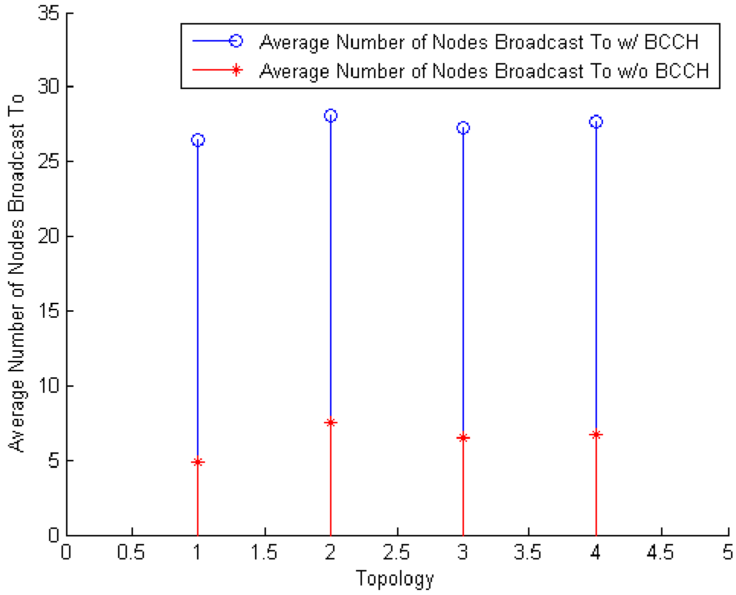


Figure 44. The average number of nodes broadcast to for all four topologies with broadcast CH (BCCH). The average number of nodes broadcast to decreases to less than ten when only the sink node's CH broadcasts.

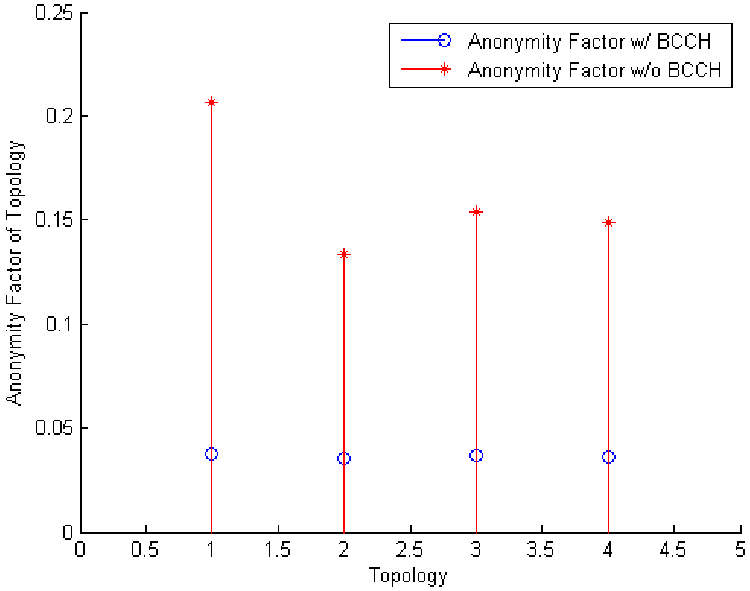


Figure 45. The anonymity factor with the broadcast CH (BCCH). The anonymity factor increases (which is a negative) without the use of broadcast CHs.

## D. CHAPTER SUMMARY

The MATLAB files and simulations that were run to evaluate the anonymity routing algorithm were reviewed in this chapter. We saw consistent results that demonstrate on a realistic network model of a WSN, we can preserve the anonymity of the sink node and without sacrificing the network lifetime.

# VII. CONCLUSIONS AND FUTURE WORK

## A. SUMMARY AND CONCLUSIONS

WSNs can be used for a variety of military, civilian and commercial applications. This thesis was motived by the proliferation of WSNs for military applications. The existing research focused on energy conservation without concern for WSN privacy or WSN privacy without concern for the limited resources of a WSN. The research in both areas failed to address realistic topologies for real world applications.

We surveyed the existing research in both the privacy and energy conservation fields to look for contributions from both fields which could be brought together to develop a routing algorithm that holistically addresses the especially vital issue of sink node privacy/anonymity in a resource efficient manner. From the energy conservation perspective, we adopted a clustering algorithm which provides energy efficient performance [3, 13, 19]. From the privacy perspective, we found concepts on dummy traffic generation and methods to define and evaluate the anonymity [5, 9, 10]. We set out to implement the algorithm, run simulations and ascertain results so that the algorithm could be evaluated for security robustness and energy preservation.

Our model and routing algorithm were implemented and simulated in MATLAB. From our simulations we were able to glean significant results. The anonymity factor is independent of traffic volume for the routing algorithm. We found that the anonymity factor varied from topology to topology and across the different simulated traffic volumes but that the results were ultimately independent of the traffic volume. The average number of nodes that were broadcast to by the broadcast CHs ranged from 21 on the low end to 32 on the high end and was consistently from 25 to 30. The average anonymity across the four topologies was 0.036566. To explain this in simpler terms, an adversary conducting traffic analysis of the deployed WSN has a less than 4% chance of finding the sink node on his/her first guess when physically searching for the sensor. This is significantly better than the conservation schemes discussed in Chapter III where all of the traffic converges on the sink node.

The average energy consumed by a node, discussed in Chapter VI, Section B5, was determined to be consistent across the four topologies in the simulations. This is promising because the average energy use by each node is an effective parameter for planning overall network lifetime. The average energy consumed by a node and the minimum energy consumed by a node both produced consistent results and increase as traffic volume increases in all cases. The maximum energy consumed by a node was variable on our simulations. Examining the results of the individual topologies, we conclude that the MaxECN plays a variety of roles within the WSN, from source node and cluster member, to cluster head, broadcast cluster head, and sink node's cluster head. In every trial, at every traffic volume, the MaxECN was the sink node's cluster head for at least one rotation. Each of these roles contributes to the energy consumption of the node, and because each of these roles are randomized from acting as source node to electing to become a cluster head, the MaxEC is highly variable.

## B. CONTRIBUTIONS OF THIS THESIS

We set out to address the issue of sink node privacy/ anonymity in a resource efficient manner. The contributions of this thesis can be summarized as follows:

- Development and implementation of a network topology and clustering algorithm in a resource—efficient manner.
- Development of a routing algorithm for sink node anonymity.
- Simulation and evaluation of the routing algorithm for security robustness and energy preservation.

To the best of our knowledge, this is the first work that develops a sink node anonymity algorithm in a resource efficient manner.

## C. FUTURE WORK

We believe that bringing these together the notion of energy efficiency and sink node privacy is vital to military applications of WSNs. The foundation to simultaneously achieve both objectives was provided by this thesis. Future work is suggested as follows.

### 1. The Maximum Energy Consumed Metric

In this thesis we found that the maximum energy consumed by a node was the most variable value of our results. We believe that one possible way to address some of these variations is to move away from nodes selecting to become CHs with a fixed probability. Some of the variations of LEACH [2,3] have different methods for choosing CHs. It is recommended that these methods of electing CHs be implemented and evaluated to see if they offer an improvement over the existing approach.

### 2. Evaluate Network Lifetime

We have determined that this routing algorithm is both privacy preserving and energy efficient, but we limited our total simulated message traffic to 20,000 messages across the WSN. Given the energy consumption trends across all four topologies, we believe that the maximum number of messages to pass through the network is much higher and that our results will scale accordingly until the point which nodes begin to die out. It is recommended that more research be conducted to determine how the algorithm performs when nodes begin to die.

### 3. Implement Routing Algorithm on a Robust Modeling Platform

We used MATLAB exclusively in this thesis. There are more robust simulation platforms that can be used. Depending on the capabilities of the platform, we can build a model that more closely aligns with the actual behavior of a sensor node. Example platforms are QualNet and OpNet.

THIS PAGE INTENTIONALLY LEFT BLANK

# APPENDIX

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Simulations Master Program %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


clear
clc
close all

% STEP 1: Generate a network and attribute initial energy values

% Note, may want to pull some parameters out of Create_RSN like max
% distance or number of nodes and initialize here. Ok as is for now.
Create_RandomSensorNetwork
Energy_Values
ElectCH
%Plot_Results

% STEP 2 : Cluster Head Cluster Member Topology
% Check CH adj matrix to make sure that no cluster heads are isolated,
% rotate cluster heads if they are.
% Create ADJ Matrix
CHadj
% Check for isolation result is vector "t" which is logical
% CH_isotest also checks the handles reelecting CH and ensuring they
are
% not isolated.
CH_isotest
% Save the Energy Values back to the original topology- will need this
% later
SaveEnergyValues

% STEP 3: CH develop routes
CH_Route
SourceSim;

% STEP 4: Route the traffic
record_num_NodesBroadcast= [];
record_SNCH_members= [];

Choose_BroadcastCH
Sim_Loop_2
Energy_Metrics
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Create_RandomSensorNetwork %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


%% Network Initialization
% number of Nodes in Network
n= 100;

% Boundaries of Network
% Sensor Field is 0-xmax by 0-ymax (square/ rectangle)
xmax=100;
ymax=100;

% N is data structure for each Node in Field
% a serves as an index

for a = 1:n
    N(a).xd= rand(1,1)*xmax;
    N(a).yd= rand(1,1)*ymax;
end

%Designate the sink node location, this assumes your sink node location
is
%deliberate
N(1).xd= 25;
N(1).yd=75;


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Energy_Values %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


% Initalize the energy in each node
% Sink node has additional resources
N(1).E= 2;

% All other nodes have fixed initial energy, Eo
Eo= .5;
for aa=2:n
    N(aa).E= Eo;
end

% Energy values for Transmit, Recieve, Sense and Computation
ETx= 50* 0.00000001;
ERx= 50* 0.00000001;
Eprocess= 5* 0.00000001;
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% ElectCH %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


%% Elect Cluster Heads
% Probability of being a cluster head (this is a place marker, there
should
% be more research on this value)
p= .20;
CountCH=0;
cluster =1;
notcluster=1;

for a= 1:n
    N(a).ch= rand(1,1);
    % E cost to gen a random #
    N(a).E= N(a).E- Eprocess;
end

%Sink node can't become a ClusterHead
N(1).ch=p+.01;

for a=1:n
    if N(a).ch <p
        CountCH = CountCH +1;
    end
end

% May not even be necessary
if CountCH <=1
    for a= 1:n
    N(a).ch= rand(1,1);
    end
end

CountCH=0;
for a=1:n
    if N(a).ch <p
        CountCH = CountCH +1;
        % Build a structure which has x and y coordinates of all
clusters
        ClusterHead(cluster).xd= N(a).xd;
        ClusterHead(cluster).yd= N(a).yd;
        ClusterHead(cluster).index=a;
        % Cost of Comparision to p
        ClusterHead(cluster).E= N(a).E- Eprocess;
        cluster= cluster+1;
    else
        ClusterMember(notcluster).xd=N(a).xd;
        ClusterMember(notcluster).yd=N(a).yd;
        ClusterMember(notcluster).index=a;
        ClusterMember(notcluster).E= N(a).E- Eprocess;
        %Cost of Comparison to p
        notcluster= notcluster+1;
    end
end

%% Nodes become ClusterMembers or ClusterHeads (additional Iterations)

% Join nodes join a Cluster, if that have not elected to be a cluster
head
% Must set a max distance (transmission range limit)
maxdistance=40;

% Distance matrix
dismatrix= zeros(cluster-1, notcluster-1);
for i=1:cluster-1
    for j =1:notcluster-1
        dismatrix(i,j)= sqrt((ClusterHead(i).xd-
ClusterMember(j).xd)^2+(ClusterHead(i).yd-ClusterMember(j).yd)^2);
    end
end

% Costs for distance matrix
for i=1:cluster-1
    ClusterHead(i).E= ClusterHead(i).E- Eprocess.*(notcluster-1);
end

for j= 1:notcluster-1
    ClusterMember(j).E=ClusterMember(j).E-Eprocess.*(cluster-1);
end

[nm, r] = min(dismatrix);

NewClusterHead1= 0;
% If in range elect to be in a Cluster, if not a portion become
ClusterHeads.
% Probablility for ClusterMembers out of Range to elect to become
% ClusterHeads as well
p2= .10;

for i=1:notcluster-1
    if nm(i)<=maxdistance
        % Don't actually have to do this, because you may reassign on
2nd
        % iteration but no harm in doing it.
        ClusterMember(i).ch=r(i);
        % Cost of Comparision
        ClusterMember(i).E= ClusterMember(i).E- Eprocess;
    else
        ClusterMember(i).ch=rand(1,1);
        % Cost of Rand # Gen
        ClusterMember(i).E= ClusterMember(i).E- Eprocess;
        % To handle sink node not becoming a ClusterHead
        ClusterMember(1).ch= p2+.01;
        % For the Case where not all ClusterMembers out of Range Elect
to
        % become ClusterHeads
        if ClusterMember(i).ch <p2
            ClusterHead(CountCH+1).xd= ClusterMember(i).xd;
            ClusterHead(CountCH+1).yd= ClusterMember(i).yd;
            ClusterHead(CountCH+1).index= ClusterMember(i).index;
            ClusterHead(CountCH+1).E= ClusterMember(i).E- Eprocess;
            CountCH=CountCH+1;
            NewClusterHead1= NewClusterHead1+1;
        end
    end
end

% To remove Self Elected ClusterHead from ClusterMember List
for i=1:notcluster-1
    m(i)=ClusterMember(i).ch>=p2;
end
notcluster= notcluster-NewClusterHead1;
cluster= cluster+NewClusterHead1;
ClusterMemberRd1= ClusterMember(m);

%Second Iteration of Associating Cluster Members to Clusterheads
% Pre-allocate Matrix
dismatrix2= zeros(CountCH, notcluster-1);

% Compute Distance's
for i=1:CountCH
    for j =1:notcluster-1
        dismatrix2(i,j)= sqrt((ClusterHead(i).xd-
ClusterMemberRd1(j).xd)^2+(ClusterHead(i).yd-
ClusterMemberRd1(j).yd)^2);
    end
end

% Costs for distance matrix
for i=1:CountCH
    ClusterHead(i).E= ClusterHead(i).E- Eprocess.*(notcluster-1);
end

for j= 1:notcluster-1
    ClusterMemberRd1(j).E=ClusterMemberRd1(j).E-Eprocess.*(CountCH);
end

NewClusterHead2=0;
[nm, r] = min(dismatrix2);

for i=1:notcluster-1
    % Nodes within Range Become ClusterMembers
    if nm(i)<=maxdistance %this is arbitrary
        ClusterMemberRd1(i).ch=r(i);
    % Nodes not within Rannge become ClusterHeads
    else
        ClusterHead(CountCH+1).xd= ClusterMemberRd1(i).xd;
        ClusterHead(CountCH+1).yd= ClusterMemberRd1(i).yd;
        ClusterHead(CountCH+1).index= ClusterMemberRd1(i).index;
        ClusterHead(CountCH+1).E= ClusterMemberRd1(i).E- Eprocess;
        CountCH=CountCH+1;
        NewClusterHead2= NewClusterHead2+1;
    end
end

% This removes Self Elected Cluster Heads (2nd Round) from
ClusterMemmber
% list
for i=1:notcluster-1
    r(i) = (ClusterMemberRd1(i).ch >=1);
end

r=logical(r);
notcluster= notcluster-NewClusterHead2;
cluster= cluster+NewClusterHead2;
ClusterMemberRd2 = ClusterMemberRd1(r);

% At this point network topoligy is complete. All nodes are placed and
all
% nodes are either Cluster Heads or Cluster Members

%% Assign a reference number to each node
start =1;
for i=1:CountCH
    ClusterHead(i).number= start;
    start=start+1;
end

counter=1;
for i=1:n-CountCH
    ClusterMemberRd2(i).number=counter;
    counter=counter+1;
end


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% CHadj %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


%Pre-allocates matrix
adjCH= zeros(CountCH,CountCH);

%Calculate distances
for i=1:CountCH
    for j =1:CountCH
        adjCH(i,j)= sqrt((ClusterHead(i).xd-
ClusterHead(j).xd)^2+(ClusterHead(i).yd-ClusterHead(j).yd)^2);
    end
end
```

```
%Reduce matrix to neighborbors within range
for i=1:CountCH;
    for j=1:CountCH;
       if adjCH(i,j)> maxdistance;
        adjCH(i,j)=0;
       end
    end
end

% Costs for adj matrix
for i=1:CountCH
    ClusterHead(i).E= ClusterHead(i).E- Eprocess.*(CountCH);
end

% Costs for Range Check
for i=1:CountCH
    ClusterHead(i).E= ClusterHead(i).E- Eprocess;
end


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% CH_isotest %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


[S,C]= graphconncomp(sparse(adjCH), 'Weak', true);

while S>1
    % Reelect CH
    SaveEnergyValues;
    clear ClusterHead
    ElectCH;
    CHadj;
    [S,C]= graphconncomp(sparse(adjCH), 'Weak', true);

end


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Dij %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


function [ShortestPath] = Dij(costs, source, dest)

n= size(costs,1);
S(1:n)=0;
distance(1:n)= inf;
previous(1:n)=inf;

distance(source) =0;
```

```
while (sum(S) ~= n)
    cand = [];
    for (i=1:n)
        if (S(i)==0)
            cand = [cand distance(i)];
        else
            cand =[cand inf];
        end
    end

    [x u] = min(cand);
    S(u) =1;

    for (i =1:n)
        if (distance(u) + costs(u,i) < distance(i)) && (costs(u,i) ~=0)
            distance(i) = distance(u) + costs(u,i);
            previous(i) = u;
        end
    end
end

distance;
previous;

ShortestPath = [dest];
traverse= dest;

while (previous(traverse) ~=source)
    ShortestPath = [ previous(traverse) ShortestPath];
    traverse = previous(traverse);
end

ShortestPath = [previous(traverse) ShortestPath];


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% CH_Route %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


a= ClusterMemberRd2(1).ch;

for i=1:CountCH
    if i==a
        ClusterHead(i).Rte= a;
    else
        ClusterHead(i).Rte= [Dij(adjCH,i,a)];
    end
end

%Energy Costs
for i= 1:CountCH
```

```
    ClusterHead(i).E= ClusterHead(i).E- ((ETx+ Eprocess+ERx)*CountCH);
end

% Count # of Route that are 2 hop (Source and SNCH)
for i=1: CountCH
    w(i)= length(ClusterHead(i).Rte);
end

Two_hop_paths= sum(w==2);
Three_hop_paths= sum (w==3);
Four_hop_paths= sum (w==4);
Five_hop_paths= sum(w==5);
SixandGreater_hop_paths= sum(w>=6);

index_two_hop= find(w==2);


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Choose_BroadcastCH %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


CHmember_Ct= zeros(1,CountCH);

% Determine how many clustermembers each CH has
for i= 1:length(ClusterMemberRd2)
    CHmember_Ct(ClusterMemberRd2(i).ch)=
CHmember_Ct(ClusterMemberRd2(i).ch) +1;
end

% Determine the Energy Value for each CH
CHenergy= [ClusterHead.E];

% Sort CH by Energy and # of CM
[sortedValuesMEM, sortedIndexMEM] = sort(CHmember_Ct(:), 'descend');
[sortedValuesE, sortedIndexE] = sort(CHenergy(:), 'descend');

totalMembers=0;
threshold_for_am= 20;
BroadCastCH= [];

i=1;
while totalMembers< threshold_for_am
    BroadCastCH(i)= sortedIndexE(i);
    totalMembers= totalMembers + CHmember_Ct(sortedIndexE(i));
    i=i+1;
end

% Add Sink Nodes CH if not already a member
SNCHcheck= ismember(ClusterMemberRd2(1).ch, BroadCastCH);

if SNCHcheck ==0
    BroadCastCH= [BroadCastCH ClusterMemberRd2(1).ch];
    totalMembers= totalMembers + CHmember_Ct(ClusterMemberRd2(1).ch);
end

Anonymity_Metrics
Two_Hop_Route_BCCH= sum(ismember(index_two_hop, BroadCastCH));
TotalBCCH= length(BroadCastCH);
TotalCH= CountCH;

BCCHList=BroadCastCH


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% SourceSim %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


% NOTE THIS IS A UNIFORM RANDOM DISTRO
Source_Node= [];
num_mess= 5000;
for d=1:num_mess %Number of messages sent through network
    Source_Node(d)= randi([2,n]); % Source node/ sink node is excluded
end


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Sim_Loop_2 %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


reset_energy= Eo/100;
reset_rds= 1000;
rds_ctr= 0;
e_tally= zeros(1, CountCH);
CH_IndexMatrix= [ClusterHead.index]
CM_IndexMatrix= [ClusterMemberRd2.index];
SNCH= ClusterMemberRd2(1).ch

for i= 1:num_mess
    if (max(e_tally) < reset_energy & rds_ctr < reset_rds)
       % Increment Counter
        rds_ctr= rds_ctr +1;
        % Route Traffic
        % Source is not a CH
        if ismember(Source_Node(i), CH_IndexMatrix) ==0
            % Route From SN to CH/ decrement energy
            indexCM(i)= find(CM_IndexMatrix== Source_Node(i),1);
            ClusterMemberRd2(indexCM(i)).E=
ClusterMemberRd2(indexCM(i)).E - Eprocess- ETx;
            CH= ClusterMemberRd2(indexCM(i)).ch;
            % Route From Source Node CH to Sink Node CH/ decrement
energy
            Rte= ClusterHead(CH).Rte;
            for j=1:length(Rte)
```

```
                % Not a Broadcast CH
                if ismember(Rte(j), BroadCastCH)== 0
                    ClusterHead(Rte(j)).E= ClusterHead(Rte(j)).E- ERx-
Eprocess- ETx;
                    e_tally(Rte(j))= e_tally(Rte(j)) + ERx+ Eprocess+
ETx;
                % Broadcast CH
                else
                    ClusterHead(Rte(j)).E= ClusterHead(Rte(j)).E- ERx-
Eprocess- ETx*CHmember_Ct(Rte(j));
                    e_tally(Rte(j))= e_tally(Rte(j)) + ERx+ Eprocess+
ETx*CHmember_Ct(Rte(j));
                end
            end
        % Source is a CH
        else
            %Route from SN to Sink Node CH/ decrement energy
            indexCH(i)= find(CH_IndexMatrix== Source_Node(i),1);
            Rte= ClusterHead(indexCH(i)).Rte;
            for k= 1:length(Rte)
                %Not a Broadcast CH
                if ismember(Rte(k), BroadCastCH)== 0
                    ClusterHead(Rte(k)).E= ClusterHead(Rte(k)).E- ERx-
Eprocess- ETx;
                    e_tally(Rte(k))= e_tally(Rte(k)) + ERx+ Eprocess+
ETx;
                % Broadcast CH
                else
                    ClusterHead(Rte(k)).E= ClusterHead(Rte(k)).E- ERx-
Eprocess- ETx*CHmember_Ct(Rte(k));
                    e_tally(Rte(k))= e_tally(Rte(k)) + ERx+ Eprocess+
ETx*CHmember_Ct(Rte(k));
                end
            end
        end
    else
        % Return Counter to 1 (Not 0 because 1 message is routed here)

        disp('**********')
        rds_ctr=1;
        % Rotate CH
        SaveEnergyValues;
        clear ClusterHead
        ElectCH;
        CHadj;
        CH_isotest;
        CH_Route;
        CHmem= [ClusterMemberRd2.index];
        CHind= [ClusterHead.index]
        SNCH= ClusterMemberRd2(1).ch
        Choose_BroadcastCH
        e_tally= zeros(1, CountCH);
        CH_IndexMatrix= [ClusterHead.index];
        CM_IndexMatrix= [ClusterMemberRd2.index];
        % Route Traffic
        % Source is not a CH
        if ismember(Source_Node(i), CH_IndexMatrix) ==0
            % Route From SN to CH/ decrement energy
            indexCM(i)= find(CM_IndexMatrix== Source_Node(i),1);
            ClusterMemberRd2(indexCM(i)).E=
ClusterMemberRd2(indexCM(i)).E - Eprocess- ETx;
            CH= ClusterMemberRd2(indexCM(i)).ch;
            % Route From Source Node CH to Sink Node CH/ decrement
energy
            Rte= ClusterHead(CH).Rte;
            for j=1:length(Rte)
                % Not a Broadcast CH
                if ismember(Rte(j), BroadCastCH)== 0
                    ClusterHead(Rte(j)).E= ClusterHead(Rte(j)).E- ERx-
Eprocess- ETx;
                    e_tally(Rte(j))= e_tally(Rte(j)) + ERx+ Eprocess+
ETx;
                % Broadcast CH
                else
                    ClusterHead(Rte(j)).E= ClusterHead(Rte(j)).E- ERx-
Eprocess- ETx*CHmember_Ct(Rte(j));
                    e_tally(Rte(j))= e_tally(Rte(j)) + ERx+ Eprocess+
ETx*CHmember_Ct(Rte(j));
                end
            end
        % Source is a CH
        else
            %Route from SN to Sink Node CH/ decrement energy
            indexCH(i)= find(CH_IndexMatrix== Source_Node(i),1);
            Rte= ClusterHead(indexCH(i)).Rte;
            for k= 1:length(Rte)
                %Not a Broadcast CH
                if ismember(Rte(k), BroadCastCH)== 0
                    ClusterHead(Rte(k)).E= ClusterHead(Rte(k)).E- ERx-
Eprocess- ETx;
                    e_tally(Rte(k))= e_tally(Rte(k)) + ERx+ Eprocess+
ETx;
                % Broadcast CH
                else
                    ClusterHead(Rte(k)).E= ClusterHead(Rte(k)).E- ERx-
Eprocess- ETx*CHmember_Ct(Rte(k));
                    e_tally(Rte(k))= e_tally(Rte(k)) + ERx+ Eprocess+
ETx*CHmember_Ct(Rte(k));
                end
            end
        end
    end
end
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Energy Metrics %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


% For Capturing Energy Consumption Data
% Return Energy Values to "N" data structure
SaveEnergyValues;
% Create "Energy Used Vector" to Capture total energy spent by each
member
EnergyUsed(1)= 2-N(1).E;
for i= 2:n
    EnergyUsed(i)= Eo- N(i).E;
end
% Avg, Max and Min Energy Used by any node and it's index
avg_EnergyUsed= mean(EnergyUsed)
[max_EnergyUsed, index_max]= max(EnergyUsed)
[min_EnergyUsed, index_min]= min(EnergyUsed)
% Number of times Max and Min E Cost Values appear in Source Node
Matrix
occurs_inSN_max= length(find(Source_Node== index_max))
occurs_inSN_min= length(find(Source_Node== index_min));


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Anonymity_Metrics %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


% For Capturing Anonymity Data
% record_num_NodesBroadcast= [];
% record_SNCH_members= [];
record_num_NodesBroadcast= [record_num_NodesBroadcast totalMembers];
record_SNCH_members= [record_SNCH_members
CHmember_Ct(ClusterMemberRd2(1).ch)];

avg_Anony= mean(record_num_NodesBroadcast);
avg_SNCH_mem= mean(record_SNCH_members);


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% SaveEnergyValues %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


% Save Energy Values Back to Original

for i= 1:CountCH
    N(ClusterHead(i).index).E= ClusterHead(i).E;
end

for i=1:n-CountCH
    N(ClusterMemberRd2(i).index).E= ClusterMemberRd2(i).E;
```

```
end


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Plot Results %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


% Plot network Topology
Network_Topology= figure;
hold on
for a=1:n
    plot(N(a).xd, N(a).yd, 'go');
    %Allows you to see Sink Node
    plot(N(1).xd, N(1).yd, 'b*');
end
hold off

axis([0 xmax 0 ymax])
xlabel('Sensor Coordinate X (meters)')
ylabel('Sensor Coordinate Y (meters)')
% title('Random Sensor Network with Deliberate Sink Node Placement')
legend('Location', 'best', 'Nodes', 'Sink Node')

% Plot new ClusterHeads
% New ClusterHeads as Red +
Iteration1= figure;
hold on
for a=1:n
    plot(N(a).xd, N(a).yd, 'go');
    plot(N(1).xd, N(1).yd, 'b*');
    for i=1:CountCH-NewClusterHead1-NewClusterHead2
        plot(ClusterHead(i).xd, ClusterHead(i).yd, 'r+')
    end
end

    %Allows you to see Sink Node

% for i=1:CountCH-NewClusterHead1-NewClusterHead2
%     plot(ClusterHead(i).xd, ClusterHead(i).yd, 'r+')
% end
axis([0 xmax 0 ymax])
xlabel('Sensor Coordinate X (meters)')
ylabel('Sensor Coordinate Y (meters)')
%title('Random Sensor Network with Deliberate Sink Node Placement and
Cluster Heads')
legend('location', 'best', 'Node', 'Sink Node', 'Cluster Head')
hold off

% Plot new ClusterHeads
% New ClusterHeads Black Diamond
Iteration2=figure;
hold on
for a=1:n
```

```
    plot(N(a).xd, N(a).yd, 'go');
    %Allows you to see Sink Node
    plot(N(1).xd, N(1).yd, 'b*');
    for i=1:CountCH-NewClusterHead1-NewClusterHead2
        plot(ClusterHead(i).xd, ClusterHead(i).yd, 'r+')
            for i=CountCH-NewClusterHead1-NewClusterHead2+1:CountCH-
NewClusterHead2
                plot(ClusterHead(i).xd, ClusterHead(i).yd, 'kd')
            end
    end

end

hold off
axis([0 xmax 0 ymax])
xlabel('Sensor Coordinate X (meters)')
ylabel('Sensor Coordinate Y (meters)')
%title('Random Sensor Network with Deliberate Sink Node Placement and
Cluster Heads')
legend('location', 'best', 'Nodes', 'Sink Node', 'Cluster Heads Rd 1',
'Cluster Heads Rd 2')

% Plot new ClusterHeads
% New ClusterHeads Red Star
Iteration3=figure;
axis([0 xmax 0 ymax])
hold on
for a=1:n
    plot(N(a).xd, N(a).yd, 'go');
    %Allows you to see Sink Node
    plot(N(1).xd, N(1).yd, 'b*');
    for i=1:CountCH-NewClusterHead1-NewClusterHead2
        plot(ClusterHead(i).xd, ClusterHead(i).yd, 'r+')
         for i=CountCH-NewClusterHead1-NewClusterHead2+1:CountCH-
NewClusterHead2
             plot(ClusterHead(i).xd, ClusterHead(i).yd, 'kd')
             for i=CountCH-NewClusterHead2+1:CountCH
                 plot(ClusterHead(i).xd, ClusterHead(i).yd, 'rp')
             end
         end
    end
end

hold off
axis([0 xmax 0 ymax])
xlabel('Sensor Coordinate X')
ylabel('Sensor Coordinate Y')
title('Random Sensor Network with Deliberate Sink Node Placement and
Cluster Heads')
legend('location', 'best', 'Nodes', 'Sink Node', 'Cluster Heads Rd 1',
' Cluster Heads Rd 2' ,'Cluster Heads Rd 3')
```

THIS PAGE INTENTIONALLY LEFT BLANK

# LIST OF REFERENCES

[1] M. Conti, “Body, personal and local ad hoc wireless networks,” in *The Handbook of Ad Hoc Wireless Networks,* M. Ilyas, Ed. Boca Raton, FL: CRC Press, 2003.

[2] J. N. Al-Karaki and A. E. Kamal, “Routing  techniques in Wireless Sensor Networks: A survey,” *IEEE Wireless Communications*, vol. 11, no. 6, pp. 6–28, Dec. 2004.

[3] A. Rahman et al., “A survey on energy efficient routing techniques in Wireless Sensor Network,” in *15th International Conference on Advanced Communications Technology*, 2013, pp. 200–205.

[4] K. A. White, “Tactical Network load balancing in multi-gateway Wireless Sensor Networks,” M.S. thesis, Department of Electrical and Computer Engineering, Naval Post Graduate School, Monterey, CA, 2013.

[5] Y. Ebrahimi and M. Younis, “Using deceptive packets to increase base station anonymity in Wireless Sensor Network,” in *Proc. Wireless Communications and Mobile Computing Conference*, 2011, pp. 842–847.

[6] M. Shao et al., “Towards statistically strong source anonymity for sensor networks,” in *Proc. IEEE Conference on Computer Communications,* 2008, pp. 466–474.

[7] N. P. Karthickraja and V. Sumathy, “A study of routing protocols and a hybrid routing protocol based on rapid spanning tree and cluster head routing in wireless sensor network,” in *Proc. IEEE International Conference on Wireless Communications and Sensor Computing,* 2010, pp. 1–6.

[8] United States Marine Corps (1997). Marine Corps Warfighting Publication 2-15.1 *Remote Sensor Operations*. Washington, DC: GAO.

[9] K. Mehta, D. Liu and M. Wright, “Protecting location privacy in sensor networks against a global eavesdropper,” *IEEE Transactions on Mobile Computing*, vol. 11, no. 2, pp. 320–336, Feb. 2012.

[10] G. Chai et al., “Enhancing sink-location privacy in wireless sensor networks through k-anonymity,” *International Journal of Distributed Sensor Networks*, vol. 2012, pp. 1-16, 2012.

[11] W. Stallings, “Data communications, data networks, and the Internet*,”* in *Data and Computer Communications,* 9th ed., Upper Saddle River, NJ: Prentice Hall, 2011.

[12] C.-H. Wu and J. D. Irwin, “An introduction to information networks,” in *Introduction to Computer Networks and Cyber Security*. Boca Raton, FL: CRC Press, 2013.

[13] I. F. Akyildiz et al., “A survey on sensor networks,” *IEEE Communications Magazine*, vol. 40, no. 8, pp. 102–114, Aug. 2002.

[14] X. Chen et al., “Sensor Network Security: A Survey,” *IEEE Communications Surveys and Tutorials,* vol. 11, no. 2, pp. 52-73, 2009.

[15] K. Mehta, D. Liu and M. Wright, “Location Privacy in Sensor Networks Against a Global Eavesdropper,” in *IEEE International Conference on Network Protocols*, 2007, pp. 313-323.

[16] Y. Jian et al., “A novel scheme for protecting receiver's location privacy in wireless sensor networks,” *IEEE Transactions on Wireless Communications*, vol. 7, no. 10, pp. 3769-3779, Oct. 2008.

[17] E. C.-H Ngai, “On providing Sink Anonymity for Sensor Networks,” in *Security and Communications* Networks, John Wiley & Sons, 2010, pp. 267-273.

[18] J. R. Ward and M. Younis, “On the use of distributed relays to increase base station anonymity in Wireless Sensor Networks,” in *Proc. IEEE Military Communications Conference*, 2012, pp. 1–6.

[19] G. Anastasi et al., “Energy conservation in wireless sensor networks: A survey,” *Ad Hoc Networks*, vol. 7, no. 3, pp. 537-568, May 2009.

[20] N. Perwaiz and M. Y. Javed, “A study on distributed diffusion and its variants,” in *12th International Conference on Computing and Information Technology*, 2009, pp. 44–49.

[21] C. Intanagonwiwat et al., “Directed diffusion: A scalable and robust communication paradigm for sensor networks,” in *Proc. ACM International Conference on Mobile Computing and Networking*, 2000, pp. 56–67.

[22] J. Kulik et al., “Negotiation-based protocols for disseminating information in Wireless Sensor Networks,” *Wireless Networks*, vol. 8, no. 2, pp. 169–185, Mar. 2002.

[23] Z. Rehena et al., “A modified SPIN for wireless sensor networks,” in *Proc. IEEE International Conference on Communications Systems and Networks*, 2011, pp. 1–4.

[24] L. Jing et al., “Energy saving routing algorithm based on SPIN protocol in WSN,” in *International Conference on Image Analysis and Signal Processing*, 2011, pp. 416–419.

[25] W. R. Heinzelman et al., “Energy-efficient communication protocol for wireless microsensor networks,” in *Proc. of the 33rd Annual Hawaii International Conference on System Sciences*, 2000, pp. 1–10.

[26] C, Hart, “Graph theory topics in computer networking,” Dept. of Comput. and Math. Sci., Univ. of Houston, 2013.

THIS PAGE INTENTIONALLY LEFT BLANK

# INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California